January 1992

Computer Software Defects: Should Computer Software Manufacturers Be Held Strictly Liable for Computer Software Defects?

Patrick T. Miyaki

Follow this and additional works at: http://digitalcommons.law.scu.edu/chtlj

Part of the Law Commons

Recommended Citation
Available at: http://digitalcommons.law.scu.edu/chtlj/vol8/iss1/4
COMMENTS

COMPUTER SOFTWARE DEFECTS: SHOULD COMPUTER SOFTWARE MANUFACTURERS BE HELD STRICTLY LIABLE FOR COMPUTER SOFTWARE DEFECTS?

Patrick T. Miyaki†

I. INTRODUCTION

As the Twentieth Century comes to a close and the new century begins, a different and more complex society evolves. Children no longer rely on board games to keep them occupied, but now spend hours on end playing Nintendo. Society no longer depends on the typewriter to draft documents, for the word processor has made such tasks much easier and more efficient. Businesses no longer have rooms full of bookkeeping materials, because all such data can be stored on computer disk. Soon law libraries will not be stocked with thousands of books because computerized legal services are flourishing. Technology, in particular the computer and the software which controls it, has now entered every facet of our lives; and as we enter the Twenty-first Century, computers are being used in more complicated areas, where the consequences of a malfunction can be both devastating and deadly.

While computers themselves are becoming more sophisticated, it is the computer software which has jumped "leaps and bounds" beyond what was ever dreamed of twenty years ago. It is this type of advanced computer software, and its defects, which has motivated the modern courts to go beyond contract liability to hold computer software manufacturers¹ liable under a computer mal-

---

Copyright © 1992 by Patrick T. Miyaki.

† B.A. 1989, Summa cum laude, San Francisco State University; J.D. Candidate, 1992, Santa Clara University School of Law.

¹ Throughout this Comment I use the term computer software "manufacturer". I use this term only to differentiate this potential defendant from a seller of software or a developer of software. I do not intend to answer the question posed later in the Comment, that the computer software manufacturer is a supplier of a product as opposed to a provider of a service, as the term "manufacturer" may imply.
practice theory. One example of an advanced computer software defect which caused great concern both in the computer services arena as well as to the public in general, occurred when AT&T's telephone network collapsed, disconnecting or causing delays in more than half of its telephone traffic. This collapse was due to a mysterious failure of a complex computer software program.

The AT&T crash caused only economic loss. However, computer software can also be used in potentially life-threatening situations. In 1985 and 1986, there were several incidents involving computer software used to monitor radiation therapy. Those incidents involved the use of the Therac 25, a computerized therapeutic radiation machine manufactured by Atomic Energy of Canada, Ltd. The Therac 25 administered to several different patients an overdose of radiation which caused either serious injury or death.

The potential damage that can result from a computer software defect is astronomical. Given the proven effects of computer software failures in the above examples, it is easy to imagine the disastrous effects that computer software defects may cause in other contexts, such as air travel. Already computer software defects have been blamed for jets flying to the wrong destination. Additionally, the prototype F-16 fighter's computer software was such that it would have caused the aircraft to flip upside down whenever it crossed the equator. The latest 747 airplane is replete with computer software; everything from the navigational system to the toilet is controlled by computer software. It seems just a matter of time before there is a mid-air collision between two computer controlled airplanes caused by a defect in the computer software.

The focus of this Comment is not on the extent of strict products liability on the computer itself, but rather the extent of strict products liability for computer software defects. This Comment

2. See Diversified Graphics, Ltd. v. Groves, 868 F.2d 293 (8th Cir. 1989). See infra Section IIA.
4. Id.
7. Id.
8. Id.
9. Rogers & Gonzales, supra note 3, at 70.
10. Id.
11. Id.
will demonstrate that a strict products liability analysis can be applied to the manufacturers of computer software. The question then arises as to whether, based on the policy reasons behind strict products liability, a computer software manufacturer should be held liable for computer software defects under a strict products liability theory. This author believes that a computer software manufacturer should not be held liable under a strict products liability theory for computer software defects.

This Comment will begin by briefly describing the theories of liability which are available to hold a computer software manufacturer liable for defects in the computer software. It will then discuss the distinction between computer software as a product and computer software as a service. The classification of computer software as either a product or a service will determine which theory of liability will be applicable. Next, Section 402A of the Restatement (Second) of Torts, as well as California's *Barker v. Lull Engineering Co.* two pronged test, will be applied to computer software defects. Finally, this Comment will discuss the policy reasons both for and against applying a strict products liability analysis to the manufacturers of computer software.

II. BACKGROUND AND HISTORY OF LIABILITY FOR COMPUTER SOFTWARE DEFECTS

Generally, computer programs contain the information which instruct the computer as to what it should do. There are two types of computer software: system software and application software. System software is the basic software which allows the computer to function in a particular fashion. System software is not designed to do any particular task, but rather to permit many different users to adapt the same computer for their particular purpose. Application software, on the other hand, is designed to allow the user to perform a certain task. The application programmer uses the utilities provided by the system software to

---

12. Kerry M.L. Smith, *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. 744 (1988). This Comment will not discuss the structure of the computer software itself, for there are other scholarly and technical articles discussing the structure of software.

13. *Id.*


15. *Id.* Examples of system software include the operating system, language translators, and various utilities.

develop a system especially designed to meet the user's needs.\textsuperscript{17} The computer software that this Comment addresses is application software, more specifically, software used with expert systems.

Expert systems are computerized systems which provide their users with advice, guidance, suggestions, decisions, or courses of action.\textsuperscript{18} These functions are intended to influence the user's behavior, and are represented, either explicitly or implicitly, as being derived from the "data and logical rules that would be used by a human being highly trained and knowledgeable in the subject matter at issue."\textsuperscript{19} Expert systems go further than providing the user with data, but actually assist in the decision making process for these systems imitate a human authority.\textsuperscript{20} While these expert systems cannot perfectly mimic human reasoning, they can match human queries to machine explanations, thereby performing deductive reasoning using a compiled knowledge base, an inference procedure and specific user-supplied data.\textsuperscript{21} The expert computer software that this Comment discusses is software that is incorporated into a device which accomplishes a certain task, as compared with an expert program which merely provides information which is gathered and analyzed by certain persons.

A. Theories of Recovery Available for Computer Software Defects

Presently, there are several recognized theories of recovery in an action for defects in computer software. First, under Article Two of the Uniform Commercial Code, if the computer software is considered to be a good, the computer software manufacturer can be held liable for injury caused by the defective computer software under a warranty theory.\textsuperscript{22} The Uniform Commercial Code controls transactions relating to goods and imposes certain implied warranties for the safety of such goods.\textsuperscript{23}

Second, under a negligence theory, a computer software manufacturer can be held liable for injury caused by defective computer

\begin{enumerate}
\item Zammit, supra note 14, at 129.
\item Id.
\item Id. at 121.
\item U.C.C. § 2-102 (1985).
\end{enumerate}
software when: (1) the manufacturer has a duty of reasonable care, (2) the computer software manufacturer has breached that duty of care, (3) there was actual and proximate causation, and (4) there was damage. Thus, if the injured consumer can show that the computer software manufacturer, in the course of producing the computer software product, breached his duty by failing to use reasonable care in the production of the software, and the resulting harm was reasonably foreseeable, the injured consumer may have a successful cause of action under a negligence theory.

Third, and most recently, one federal court has applied a malpractice theory of liability to computer software professionals. Computer malpractice is similar to a negligence theory, except under a malpractice theory a computer professional will be held to the high standard of care equal to those practicing in the same profession. A computer software manufacturer should be held to be a computer professional because of the manufacturer's expertise in the field of computer software. Thus, if a computer software manufacturer will be held to a higher standard of care than the ordinary reasonable person under a computer malpractice theory, it will be more difficult for the computer software manufacturer to show that the standard of care has been met, thus making it easier for the consumer to recover.

The final theory of recovery available for products liability is strict liability. However, the courts have not yet held a computer software manufacturer strictly liable for computer software defects.

24. Data Processing Services, Inc. v. L.H. Smith Oil Corp., 492 N.E.2d 314 (Ind. Ct. App. 1986). A software manufacturer may be negligent in two different ways: the manufacturer may be negligent in the design of the product, and the manufacturer may be negligent in the provision of services.
27. Id. at 296.
28. Although the U.S. Court of Appeals in Diversified Graphics Ltd. never used the term "computer malpractice", it is clear that the Eighth Circuit held implicitly that the tort of computer malpractice exists. The court in Diversified Graphics Ltd. stated that the expert witnesses called by both plaintiff and defendant established sufficient evidence from which the jury could find that the accounting firm did not exercise due professional care in providing its computer consulting services. Marc S. Friedman, "Computer Malpractice": A Theory of Recovery Emerges After a Decade of Debate, THE COMPUTER LAWYER, Nov. 1989, at 14.
III. STRICT PRODUCTS LIABILITY FOR COMPUTER SOFTWARE DEFECTS

A. Computer Software as a Product Versus Computer Software as a Service

The "model" basis for strict products liability is Section 402A of the Restatement (Second) of Torts. A preliminary question that must be answered before strict liability may be imposed under Section 402A is whether computer software is a service or a product. If computer software is deemed to be a service, then a manufacturer is generally not liable in tort absent some negligent behavior or intentional misconduct. If computer software is deemed to be a product, then a manufacturer may be liable under warranty, negligent design and strict liability.

As in computer malpractice, a computer software manufacturer may be considered a provider of professional services who sells products within the authorized scope of his professional practice. Such would be the case if the computer software manufacturer develops customized computer software for a particular user, personally enters it into the hardware, and contracts to maintain it in working order.

Furthermore, one definition of a service is that it is "something which is rarely duplicated, allowing little chance of quality control or defect testing." With computer software, especially complex expert systems, it is impossible to test all the possible combinations of commands to make sure that each computer program is defect free. For example, the AT&T crash occurred when an unforeseen and undeterminable combination of telephone calls caused a computer software defect to surface in the computer software that determines the most efficient path for routing long distance telephone calls. Moreover, since one federal court has applied the computer malpractice theory for a computer software defect situation, there is an argument that implicitly the courts, at least in some circumstances, will deem the computer software manufacturer as provid-

29. Several states apply § 402A directly, and many other states use parts of § 402A or have in some way modified it.
32. Id.
34. Interview with Dr. H. Alkhatib, Professor of Computer Engineering, Santa Clara University, in Santa Clara, CA (Nov. 1, 1990).
35. Rogers and Gonzales, supra note 3, at 70.
ing a service, rather than a product. Such an interpretation would prohibit strict products liability from being applied to a computer software manufacturer.

However, with the use of computer software increasing dramatically, the marketing, packaging, and distributing of computer software has changed.\(^{36}\) Presently, it is less likely that a computer software manufacturer will develop a computer program for only one particular purpose.\(^{37}\) The computer program may be a modification of a previously developed program or a computer program developed to accomplish one task, but which may be distributed to different consumers.\(^{38}\) In these cases, where the computer software resembles a product in terms of its marketing, packaging, and volume of sales, it is more likely that the computer software will be considered a product.\(^{39}\) One federal court has suggested that a computer software program that fails to yield the result for which it was designed may be a product for purposes of products liability law.\(^{40}\)

By way of analogy, the Uniform Commercial Code broadly defines "goods" as "all things (including specially manufactured goods) which are moveable at the time of identification to the contract for sale other than money in which the price is paid, investment securities and things in action."\(^{41}\) Thus, in determining whether the Uniform Commercial Code is applicable, the emphasis is on whether the transfer of property or the provision of a service is the focus of the contract.\(^{42}\) If the focus of the contract is the transfer of property, the Uniform Commercial Code will apply.

Because the determination of whether a particular situation involves the provision of a service or the sale of a product is a crucial factor in deciding which theory of liability is available, in certain "hybrid enterprises" legislation has defined the enterprise as providing a service.\(^{43}\) These hybrid enterprises involve a combination of

---

38. Id.
39. Id.
40. Winter v. G. P. Putnam's Sons, 938 F.2d 1033 (9th Cir. 1991). This is a case in which the purchasers of a book containing information on the collection of mushrooms were injured in collecting and eating certain poisonous mushrooms. The plaintiffs had relied on information provided in the book, and they sued the publisher of the book under a products liability theory.
42. Smith, supra note 12, at 747-748.
the sale of a product and the performance of a service.\textsuperscript{44} For example, a local pharmacist's function has been legislatively defined as a profession involving a "dynamic patient-oriented health service that applies a scientific body of knowledge to improve and promote patient health by means of appropriate drug use and drug related therapy."\textsuperscript{45} Similarly, legislation has declared that the "distribution or use of whole blood or plasma is a service for all purposes and shall not be construed to be a sale."\textsuperscript{46}

Looking into the future, it is foreseeable that computer software will be used in almost every conceivable machine. The Legislature, if it so desires, could shield computer software manufacturing by defining it as a provision of a service. The Legislature could justify such a statute by claiming that it is not in the public interest to subject computer software manufacturers to strict products liability because the need for innovative, technologically advanced computer software outweighs the advantage to the individual injured party of being able to recover for injuries on a strict products liability basis.\textsuperscript{47}

\textbf{B. Manufacturing Defects Versus Design Defects}

There are three types of defects that can give rise to strict products liability: warning defects, manufacturing defects and design defects.\textsuperscript{48} This Comment will address only manufacturing and design defects. A manufacturing defect is one that differs from the intended result of the manufacturer or from other identical items of the same production line.\textsuperscript{49} A manufacturing defect, therefore, is basically a production flaw, an imperfection in the manufacturing process such that the product is not completed as designed.

A design defect, however, cannot be identified by merely comparing the product which caused the injury with the manufacturer's plans or with other items of the same production line, for the plans and all items from the same production line will reflect the same

\textsuperscript{44} \textit{Id}. In \textit{Murphy}, the California Supreme Court was considering the duties of a community pharmacist who fills prescriptions for drugs and who uses due care in preparing and labelling the drug. The \textit{Murphy} Court recognized that the local pharmacist provides a service to the physician and acts as an extension of the physician, similar to the role of a technician who analyzes a blood sample for a doctor. At the same time, the pharmacist is "in the business of selling prescription drugs, and his role begins and ends with the sale."

\textsuperscript{45} \textit{Cal. Bus. & Prof. Code} § 4046(b).

\textsuperscript{46} \textit{Cal. Health & Safety Code} § 1606.

\textsuperscript{47} See infra §§ IV & V.

\textsuperscript{48} \textit{Barker v. Lull Engineering Co.}, 573 P.2d 443, 454 (Cal. 1975); \textit{Anderson v. Owens-Corning Fiberglas Corp.}, 810 P.2d 549 (Cal. 1991).

\textsuperscript{49} \textit{Barker v. Lull Engineering Co.}, 573 P.2d 443, 454 (Cal. 1975).
Therefore, a design defect exists where the product is designed in such a way that it carries an inherent risk of harm in normal use.\textsuperscript{51}

Strict products liability is equally applicable to both manufacturing defects and design defects, although it is often easier to apply strict products liability to a manufacturing defect. The reason it is easier to apply strict products liability to a manufacturing defect is because "when an end product is more dangerous than contemplated by its design, due to a flaw in the production process the product is unreasonably dangerous."\textsuperscript{52} Also, it is easier to prove a manufacturing defect exists as opposed to a defect in design.\textsuperscript{53} If there is a manufacturing defect, the injured user only needs to prove that the defect existed.\textsuperscript{54} But if there is a design defect, the injured consumer must prove both the existence of the alleged defect and that there existed a feasible alternative to the manufacturer's selected design.\textsuperscript{55}

C. Application of Section 402A of the Restatement (Second)
of Torts to Manufacturers of Computer Software

Many jurisdictions have adopted the Restatement formulation for strict products liability either in whole or with certain modifications. The text of Section 402A of the Restatement (Second) of Torts provides:

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 (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

\textsuperscript{50} Id.
\textsuperscript{51} The definition of a design defect may vary depending upon which jurisdiction one is in and may also depend upon the factual situation involved.
\textsuperscript{52} Massingale & Borthick, \textit{supra} note 6, at 195.
\textsuperscript{53} Gill, \textit{supra} note 21, at 501.
\textsuperscript{54} Id.
\textsuperscript{55} Id.
or entered into any contractual relation with the seller.\textsuperscript{56}

In applying Section 402A to computer software, it must be divided into its elements. First, Section 402A applies to "one who sells."\textsuperscript{57} "One who sells" has been defined as applying to "any person engaged in the business of selling products for use or consumption."\textsuperscript{58} A computer software manufacturer is creating and developing software for the purpose of selling it, for if the computer programs which the company develops are not successful, they will not sell and the manufacturer will soon be out of business. Therefore, because "one who sells" is so broadly defined, a computer software manufacturer may be considered as "one who sells."

Defining a computer software manufacturer as "one who sells" for purposes of Section 402A is supportable by the overall rationale for this requirement in the Restatement. The reason for making "one who sells" specially responsible for defects in a product is twofold: (1) the manufacturer has entered into the business of supplying products which may injure persons or property, and (2) those who purchase such goods are forced to rely upon the manufacturer's undertaking.\textsuperscript{59} Thus, it can be said that many completed programs which are distributed to more than one client often have sufficient characteristics of a product to fall within strict products liability principles.\textsuperscript{60} In addition, given that these products can be dangerous and that consumers will rely upon these products, the computer software manufacturer should be held liable for putting this "product" on the market.\textsuperscript{61}

Under Section 402A, the product must be in a defective condition unreasonably dangerous to the user or the consumer or his property.\textsuperscript{62} The product is defective if "at the time it leaves the seller's hands, [it is] in a condition not contemplated by the ultimate consumer, [and] which will be unreasonably dangerous to him."\textsuperscript{63} This requirement may cause some difficulties when applied to computer software manufacturers because typically the computer software has been thoroughly tested by a quality control group

\begin{itemize}
  \item \textsuperscript{56} Restatement (Second) of Torts § 402A (1966).
  \item \textsuperscript{57} Id.
  \item \textsuperscript{58} Id., cmt. f.
  \item \textsuperscript{59} Id.
  \item \textsuperscript{60} \textsc{Raymond T. Nimmer}, \textsc{The Law of Computer Technology} § 7:06[2][b] (1985).
  \item \textsuperscript{61} For § 402A to apply, the computer software must be deemed a product, not a service. \textit{See} discussion § IIIA, \textit{supra}.
  \item \textsuperscript{62} Restatement (Second) of Torts § 402A (1966).
  \item \textsuperscript{63} Id., cmt. g.
\end{itemize}
within the manufacturing entity itself. This quality control group will test sequences to verify proper procedures and to make sure that there are no problems with the program. The ultimate consumer generally contemplates that the computer software manufacturer has tested the computer software to the best of that manufacturer's ability.

However, the quality control group is severely limited in the amount of tests that can be run on the computer software. Many programs, especially expert programs, contain thousands of instructions with millions of possible combinations. A combination of commands not considered or tested by the programmer may cause a defect in the computer program which may ultimately lead to a dangerous malfunction. It would be unrealistic and infeasible for a quality control group to be able to test each and every combination, and thus the question arises as to whether a computer software program can ever be deemed to be not in a defective condition unreasonably dangerous to the user or to the consumer. An example is the computer software involved in the AT&T crash in January 1990. The software had been thoroughly tested, but a unique combination of commands not tested caused the system to collapse. A complex computer program can be considered thoroughly and completely tested only by actually attempting every combination of commands to determine whether a failure in the computer software will surface. In some of the larger computer programs, it would take tens, if not hundreds, of years to complete this testing.

Under Section 402A, "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 the product's characteristics." This requirement may also be a limiting factor to the applicability of strict products liability to computer software programs. But in regard to computer software which can ultimately result in some type of personal injury (as in the Therac 25 case), this requirement will not preclude strict products liability given that the ordinary consumer's expectation of computer software is that the computer software is

---

64. Interview with Dr. H. Alkhatib, Professor of Computer Engineering, Santa Clara University, in Santa Clara, CA (Nov. 1, 1990).
65. Id.
66. GEMIGNANI, supra note 31, at 448.
67. Rogers and Gonzalez, supra note 3, at 70-71.
68. Id., referring to an interview with John Shen, a computer researcher at Carnegie-Mellon University.
69. Restatement (Second) of Torts § 402A cmt. i (1966).
not in such a defective condition that it will cause injury or death.\footnote{70}{Expert computer software programs are extremely complex, and in certain uses, such as when the computer software is used to monitor and treat patients, the learned intermediary doctrine may be applicable. Under this doctrine, it is the expectations of the intermediary that are considered, and not the expectations of the ultimate consumer.}

If a product is dangerous to use, but may become safe to use by adequate warnings or directions, then it is not unreasonably dangerous under Section 402A.\footnote{71}{Restatement (Second) of Torts § 402A, cmt. j (1966).} One may be tempted to analogize computer software with new drug technology and allow warnings or directions to be placed upon the computer software program as to the program's use in order to exclude the computer software from the unreasonably dangerous category.

This analogy between computer software and new drug technology is unworkable for two reasons. First, a warning need only be given if the risk of injury can be substantiated.\footnote{72}{Anderson v. Owens-Corning Fiberglas Corp., 810 P.2d 549, 558 (Cal. 1991).} In new drug situations, the risk of a certain side effect or reaction to that drug is known.\footnote{73}{It can be calculated that certain types of people may react in a certain way to a particular drug. Furthermore, it can be determined that certain types of drugs react a certain way when combined with other consumables and/or drugs. For example, Elocon Ointment is a prescription drug used to treat dermatitis (rashes). Elocon Ointment is packaged with an information sheet which warns potential users that this particular product has caused burning, puritus, skin atrophy, and furunculosis in several users of this medication.} However, with computer software, the risk is unknown due to the infinite number of combinations that must be tested in order to conclusively determine that the computer software is defect free. Therefore, when a computer software manufacturer first puts a new computer program on the market, at that time, the manufacturer cannot know with any degree of certainty that a specific combination of commands will cause the system to fail.\footnote{74}{For if the manufacturer knows that a certain sequencing will cause the software to fail, he will fix it before it is marketed for these defects can always be corrected. This is another distinction between drugs and software; drugs will unavoidably cause some side effects to some people and this cannot be corrected. Interview with Dr. H. Alkhatib, Professor of Computer Engineering, Santa Clara University, in Santa Clara, CA (Nov. 1, 1990).} It is unrealistic and illogical for a computer software manufacturer to warn about a risk that may not even exist.\footnote{75}{In California, the issue of strict products liability actions based on failure to warn has been conclusively decided. The California Supreme Court has held that knowledge, actual or constructive, is a requisite for strict liability to be imposed in failure to warn cases, even in cases other than prescription drug cases. However, this issue has not been decided in strict products liability cases based on manufacturing defects or design defects. Anderson v. Owens-Corning Fiberglas Corp., 810 P.2d 549, 557 (Cal. 1991). The focus of this Comment is on manufacturing and/or design defect cases.}
be adequate, to whom must the computer software manufacturer give a warning. The general rule is that the manufacturer must warn all persons who will foreseeable come in contact with, and consequently be endangered by, that product.\textsuperscript{76} The only exception to this rule is when the warning has been given to a responsible intermediary such that the manufacturer has no duty to directly warn the consumer.\textsuperscript{77}

The rule can easily be applied in some situations, such as with computer software used to monitor a patient's treatment, as in the Therac 25 case. This is due to the fact that the manufacturer of the computer software can warn the hospital, who can warn the doctor, who in turn can warn the patient. The problem becomes more troubling with computer software responsible for such things as the navigational system of an aircraft. The manufacturer can warn the airline and the airline can warn its personnel; but must that airline warn every passenger who buys an airline ticket? Realistically, this does not seem like a workable solution.

A way under Section 402A in which a computer software manufacturer can escape strict products liability for computer software defects is if the computer program is considered an unavoidably unsafe product.\textsuperscript{78} There are some products which, "in the present state of human knowledge, are quite incapable of being made safe for their intended and ordinary use."\textsuperscript{79} As described thus far, it appears that computer software programs may fall within this category. But, comment k of Section 402A goes on to add that such a product must be accompanied with proper directions or warnings,\textsuperscript{80} which brings up the problem previously discussed.

Therefore, in applying an analysis of Section 402A to computer software, it appears that a computer software manufacturer is a likely candidate for being held liable under a strict products liability theory. A manufacturer who sells a software program, wherein a combination of commands causes the computer software to fail, is selling a product in a defective condition unreasonably dangerous to the user or consumer or his property. Whether a particular computer software manufacturer will be held liable under Section 402A will depend on the particular facts of each case. This analysis only


\textsuperscript{77} Restatement (Second) of Torts § 388 cmt. n (1966).

\textsuperscript{78} Restatement (Second) of Torts § 402A cmt. k (1966).

\textsuperscript{79} Id.

\textsuperscript{80} Id.
concludes that in certain circumstances a computer software manufacturer may be held strictly liable under Section 402A of the Restatement (Second) of Torts, and is not precluded as such merely because the product involved is a computer software program.

1. State of the Art Defense to the Application of Section 402A of the Restatement (Second) of Torts for Manufacturers of Computer Software

The primary defense available to a manufacturer of computer software is the "state of the art" defense. The state of the art defense refers to the existing level of scientific knowledge and technological expertise relevant to a certain industry at the time a product is designed. This defense is available where the defendant did not know and could not have known of the hazards at the time the product was manufactured and marketed. State of the art may be used as an affirmative defense, and as such the manufacturer has the burden of proof to demonstrate that he could not have known about the risk involved. Depending upon which jurisdiction one is in, state of the art evidence may be useful to determine the expectations of the ordinary consumer or the feasibility of a proposed alternative design under a risk-utility test.

A computer software manufacturer may use the state of the art defense by demonstrating that after thorough testing, the manufacturer did not and could not know of the hazards that arose to cause the injury. The manufacturer may demonstrate that there are an enormous number of combinations of commands that could be tested and that the quality control group tested an amount of combinations of commands which conformed to the expertise and scientific knowledge at the time the computer software was designed.

Proof of state of the art need not, as a matter of law, compel judgment for the defendant. One may argue that although the computer software manufacturer may not have been aware, nor could have been aware, of the particular risk involved, the manufacturer did know that a risk of some injury existed based on the fact

82. Id.
83. Id.
84. See Birnbaum, supra note 33, at 154.
that all of the possible combinations had not been tested to see whether a certain combination would cause the software to fail.87

Thus, the state of the art defense may be applicable to computer software manufacturers, but this defense's applicability will depend on the amount of testing performed on the computer software. With expert computer programs, there will usually be some degree of risk that an unknown combination of commands will cause the computer software to fail.88 The state of the art defense, however, will only preclude a finding of liability when the computer software manufacturer can demonstrate that they tested an amount of combinations of commands that conformed with the standards accepted within that area of expertise.89

D. California's Strict Products Liability Law

California has a two-tiered test for the application of strict products liability to hold the manufacturer of products liable. The California Supreme Court has concluded that a product is defective in design either: (1) if the product has failed to perform as safely as an ordinary consumer would expect when used in an intended or reasonably foreseeable manner, or (2) if, in light of particular relevant factors, the benefits of the challenged design do not outweigh the risk of danger inherent in such design.90

The first tier of California's strict products liability test is basically an application of Section 402A of the Restatement (Second) of Torts as discussed supra. However, if the first tier is not satisfied, then the California courts will apply the second tier to determine whether the manufacturer may be held strictly liable.91

The second tier of the California test was developed in Barker v. Lull Engineering Co. and is a risk-utility test.92 The risk-utility test is similar to a negligence standard but the emphasis in the Barker test is on the product itself, whereas in a negligence analysis, the focus is on the manufacturer.93 The California Supreme Court determined that if, through hindsight, the trier of fact finds that the

87. In some jurisdictions, the courts will impute knowledge of the risk to the manufacturer, and thus, the manufacturer will be assumed to have known about the risk. Phillips v. Kimwood Machine Co., 525 P.2d 1033, 1037 (Or. 1974).
88. GEMIGNANI, supra note 31, at 425.
89. See id. at 448-450. At the present time, these standards may be difficult to determine.
91. Id.
92. Id.
product's design embodies an excessive preventable danger, or, in other words, if the fact finder determines that the risk of danger inherent in the challenged design outweighs the benefits of the design, the product may be found defective in design.\textsuperscript{94}

In a risk-utility analysis, once the plaintiff shows that the design caused the injury, the burden shifts to the manufacturer to demonstrate that the computer software does not embody an excessive preventable danger.\textsuperscript{95} The computer software manufacturer must show that the benefits of the computer program outweigh the risk of injury.\textsuperscript{96} The trier of fact may consider several factors in determining if the benefits do indeed outweigh the risk, including: "the gravity of danger posed by the challenged design, the likelihood that such danger would occur, the mechanical feasibility of a safer alternative design, the financial cost of an improved design, and the adverse consequences to the product and to the consumer that would result from the alternative design."\textsuperscript{97}

The problems posed by defects in computer software will most likely be addressed under the first tier of the \textit{Barker} test. But assuming there is a computer software program which performs up to the ordinary consumer expectation under the first prong of the \textit{Barker} test, the manufacturer of that program may still be held strictly liable under the risk-utility prong. For example, suppose there is a "cutting edge" advance in computer software technology and one computer software manufacturer develops a computer program that monitors AIDS patients and assists in their treatment. Because this would be such a new and experimental computer program, the ordinary consumer expectation may be very low, and therefore this computer software program would meet the threshold level needed to satisfy a Section 402A analysis.

In looking at the computer software through hindsight, however, the trier of fact might find the computer software manufacturer liable under the risk-utility prong of the \textit{Barker} test. For example, the fact finder, in retrospect, may determine that the gravity of the danger posed by the computer program was great, the

\textsuperscript{94} Barker v. Lull Engineering Co., 573 P.2d 443, 454 (Cal. 1978). The California Supreme Court, throughout the \textit{Barker} decision, refers to design defects. But, in its dicta, the Court determined that the two-tiered analysis for design defects applies equally to manufacturing defects. However, the two-tiered analysis is more pronounced in design defect cases in which the manufacturer can frequently argue that its product satisfied ordinary consumer expectations since the product was identical to other items of the same product line with which the consumer may well have been familiar.

\textsuperscript{95} \textit{Id.}

\textsuperscript{96} \textit{Id.}

\textsuperscript{97} \textit{Id.} at 431.
likelihood that the danger would occur was substantial, that it would have been feasible for the manufacturer to use a safer design and that the alternative design would not have adverse consequences to the patient. In such a case, the fact finder can, under a risk-utility analysis, hold the computer software manufacturer strictly liable.

IV. Policy Reasons for Extending Strict Products Liability to Computer Software Manufacturers

There are numerous policies behind strict products liability which would support an extension of strict liability to the manufacturer of computer software when a defect in the software causes injury.\textsuperscript{98}

A. Computer Software Defects Can Now Cause Personal Injury

First and foremost, the computer software industry has advanced rapidly in the last few years. Previous computer software defects tended to result only in economic loss, not personal injury.\textsuperscript{99} While the economic loss may have been great, the courts were content to hold the manufacturer liable on a warranty theory under the Uniform Commercial Code. Presently, however, with the sophistication of modern software in relatively new areas, the risk of personal injury is very real and the extent of that injury can be quite staggering.\textsuperscript{100} Since strict products liability tends to be used when personal injury occurs, it seems only logical now, with personal injury as a resulting risk of defective computer software, that strict products liability extend to computer software.

B. Risk Spreading Is Desirable

Strict liability allows the consumers of the product to spread the risk of loss from a defective product more equitably than under a negligence theory.\textsuperscript{101} Risk spreading attempts to disperse the burden more evenly across a wide section of society, rather than placing the burden of the cost of the injury on any one individual.\textsuperscript{102}

\textsuperscript{98} This is not intended to be an all-inclusive list of the policy reasons for imposing and for not imposing strict products liability upon computer software manufacturers. Instead, the focus is on the most important policy reasons as evaluated by this author.

\textsuperscript{99} See Massingale & Borthick, supra note 6.

\textsuperscript{100} Id.

\textsuperscript{101} GEMIGNANI, supra note 31, at 419.

\textsuperscript{102} Birnbaum, supra note 33, at 141.
Therefore, strict liability based on a risk spreading rationale can be extended to computer software manufacturers because one can predict that any computer software program may contain a defect which can potentially lead to a system failure.\textsuperscript{103} Since computer software is useful despite its potential to do harm or inflict injury if defective, the manufacturer is encouraged to purchase products liability insurance, thus passing the cost of the insurance onto the users of the product.\textsuperscript{104}

This risk spreading rationale assumes that products liability insurance will be available. In purchasing products liability insurance, the manufacturer seeks to exchange an uncertain risk, that of potential future actionable incidents involving its products, for a certain cost.\textsuperscript{105} Liability insurance does not represent insurance against the unknown, but is an offering of insurance and setting of premiums for liability based upon "the carrier's actuarial projection of what the insured's overall loses are expected to be."\textsuperscript{106} Thus, for a risk to be insurable, the risk must be (1) specified or, at least capable of being identified, and (2) the duration of the risk must be fixed and able to be determined.\textsuperscript{107}

Thus, acquiring products liability insurance may be difficult for computer software manufacturers. First, as discussed previously, the computer software manufacturer may not be able to determine the risk involved in the computer software because all the sequencing combinations cannot be tested before the computer software is marketed, especially in a complicated and complex program. Second, the risk often cannot be identified\textsuperscript{108} (and if it could, it should be eliminated). Third, the duration of the risk is not fixed given that the specific combination of commands which will cause the system to crash may never arise, or that combination of commands may only arise sometime in the distant future.\textsuperscript{109} Finally, if products liability insurance were available to computer software manufacturers, the price would be extremely high for complex programs because the gravity of injury is so great.

C. Manufacturer Makes a Representation of Product Safety

Another policy reason for strict products liability is the im-

\textsuperscript{103} Gemignani, supra note 31, at 420.
\textsuperscript{104} Birnbaum, supra note 33, at 141.
\textsuperscript{105} 2 M. Stuart Madden, Products Liability 414 (2d ed. 1988).
\textsuperscript{106} Id.
\textsuperscript{107} Id.
\textsuperscript{108} Gemignani, supra note 31 at 425.
\textsuperscript{109} See id.
licit representation of safety that the manufacturer makes when he or she places the product on the market.\textsuperscript{110} Implicit in a product's presence on the market is a representation that the product will safely do the jobs for which it was built. By placing the computer software on the market, the computer software manufacturer should be forced to assume liability for any injury caused by the computer software which was unreasonably dangerous.\textsuperscript{111} The public should receive maximum legal protection for product defects against which the public is helpless from protecting itself.\textsuperscript{112}

On the other hand, the computer software manufacturers may be unable to protect themselves against defects in their computer software. Manufacturers of software will often be unaware of a defect in their product until all the sequencing combinations are tested.\textsuperscript{113} This testing of command combinations may not feasibly be completed before the software is marketed.\textsuperscript{114}

D. Liability Under a Negligence Theory May be Unavailable

There is another policy reason favoring strict liability for manufacturers of defective computer software. The computer software manufacturer may not be able to be held liable under a negligence theory.\textsuperscript{115} The injured party, in a negligence action, faces the burden of showing that there was a lack of due care in the design or manufacture of the computer software.\textsuperscript{116} Additionally, in the more complex and advanced computer software, it may be impossible or extremely difficult for a consumer or user of the computer software, who knows little about the workings of the software, to identify the lack of due care by the manufacturer which was responsible for the defect.\textsuperscript{117}

The Eighth Circuit Federal Court of Appeals has imposed a higher standard of care upon computer software professionals because of the difficulty of proof in negligence actions.\textsuperscript{118} By applying a strict products liability theory, the courts would take the final step of extending liability to a computer software manufacturer even

\begin{itemize}
  \item \textsuperscript{110} William Prosser, Handbook of the Law of Torts 651 (1971).
  \item \textsuperscript{111} Id.
  \item \textsuperscript{112} Birnbaum, supra note 33, at 142.
  \item \textsuperscript{113} Interview with Dr. H. Alkhatib, Professor of Computer Engineering, Santa Clara University, in Santa Clara, CA (Nov. 1, 1990).
  \item \textsuperscript{114} Gemignani, supra note 31, at 425.
  \item \textsuperscript{115} Id. at 418.
  \item \textsuperscript{116} Id.
  \item \textsuperscript{117} Birnbaum, supra note 33, at 142. An injured consumer may wish to consider the application of \textit{res ipa loquitur}, but such an analysis is outside the scope of this Comment.
  \item \textsuperscript{118} Diversified Graphics, Ltd. v. Groves, 868 F.2d 293 (8th Cir. 1989).
\end{itemize}
though that manufacturer "has exercised all possible care in the preparation and sale of his product." 119

E. Privity is Not Required Under Strict Liability

The extending of strict products liability to computer software avoids the privity that is required in order to recover under the Uniform Commercial Code. 120 Strict liability applies even though the user or consumer may not have purchased the product from or entered into any contractual relation with the manufacturer. 121 Thus, strict products liability avoids the multiple litigation which occurs in warranty actions where the injured party sues the seller, the seller in turn sues the distributor, and the distributor sues the manufacturer.

The privity requirement, however, may not cause as many problems when it comes to complex computer software as it may cause with other products, such as automobiles. The manufacturer and developer of the computer software program are essentially one entity when it comes to creating complex, expert computer software. 122 The software developers in more complex programs will exist within the manufacturing entity and thus, the manufacturer will be liable for all computer programs created by its developers. 123 Often, with expert programs, the manufacturer will also be the distributor, thereby eliminating another link in the privity chain. 124 Thus, where the computer software manufacturer directly sells to the consumer a computer program which it developed within its own manufacturing entity, there is no privity problem and the injured party can recover directly from the computer software manufacturer.

F. Strict Liability Forces Manufacturers to Take Precautions Before Marketing Their Product

Another policy reason given for the implementation of strict

---

120. U.C.C. § 2-313. One must be cautioned, however, that jurisdictions differ as to whether privity is required. In certain jurisdictions it will depend upon whether the action is under an implied warranty theory or under an express warranty theory.
121. Id.
122. There may, however, be a distributor between the manufacturer and developer when it comes to the less complex software such as games. In this situation the individual developer may sell his rights to the program to a manufacturer and the privity problem arises. Interview with Dr. H. Alkhatib, Professor of Computer Engineering, Santa Clara University, in Santa Clara, CA (Nov. 1, 1990).
123. Id.
124. Id.
products liability for manufacturers of computer software is that forcing manufacturers to pay for personal injuries created by computer software defects that make the computer program dangerous will encourage manufacturers to be careful. Generally, since the manufacturer is in the most advantageous position to detect and prevent defects in their product, the manufacturer should be motivated to do so.

This policy rationale might not be applicable to the manufacturers of computer software. Strict liability may encourage computer software manufacturers to take more precautions in producing their programs, but there is "no reason to believe that any amount of care will ever guarantee a risk free program." In fact, even without the imposition of strict products liability for software defects, computer software manufacturers are going to take as many precautions as are available to protect against defects. Software manufacturers are encouraged, and for the most part do, via their quality control groups, test as many sequences and combinations as they can before marketing the computer program. The problem continually arises that it is impossible, with complex and complicated programs, to test all the available combinations of commands without taking many years.

G. Injured Party Will be Adequately Compensated

Finally, the argument is set forth that the application of strict liability to computer software defects would ensure that injured plaintiffs receive adequate compensation for their injuries. Manufacturers tend to hold enough assets to be able to compensate the injured victim and, presuming that some sort of computer software liability insurance may be available, that the manufacturer will hold a policy which would cover this type of injury.

125. GEMIGNANI, supra note 31, at 420.
126. Birnbaum, supra note 33, at 143.
127. GEMIGNANI, supra note 31, at 420.
128. Interview with Dr. H. Alkhatib, Professor of Computer Engineering, Santa Clara University, in Santa Clara, CA (Nov. 1, 1990).
129. Id.
130. See GEMIGNANI, supra note 31, at 425.
131. Birnbaum, supra note 33, at 143.
132. In addition to the problem of the availability of product liability insurance for computer software manufacturers, there is the problem of, assuming insurance is available, for what injuries the insurance policy will protect against and compensate.
V. **Policy Reasons Against Extending Strict Products Liability to Computer Software Manufacturers**

A. *Imposition of Strict Liability Will Cause an Undue Burden Upon Computer Software Manufacturers*

The extension of strict products liability to computer software manufacturers may impose an undue burden. The failure of software can be quite extensive, subjecting manufacturers to strict products liability may impose an undue burden on the computer software industry. Thus, it is essential that the need for the computer program be balanced against the ultimate danger involved if the computer software were to malfunction.

The balancing of the need versus the harm may lead to different results depending on the particular application of the computer program. For instance, where the computer software is used for the treatment or monitoring of patients, the need for an innovative computer program may be so demanding that even a high probability of a software failure will not affect the balance in favor of applying strict products liability. But, the harm may outweigh the benefit when other computer programs are considered. For example, if a computer program that is used to control the navigational systems of airplanes fails and several hundred people die as a result of the airplane crash, then it is arguable that this risk of harm outweighs the benefit of the computer software. In balancing the need versus the harm, it is crucial to look at the risk of damage had the computer program not been used and if there are other proven methods of carrying out the same function.

B. *Extent of Liability May be Difficult to Determine*

Another problem with imposing strict liability upon software manufacturers for computer software defects is determining the extent of liability. Because expert programs are so complex, it would be unfair to hold the seller of the program liable for the defects in the computer software, unless the seller were the manufacturer who developed the program. But, if the manufacturer acted as the

---

133. Gemignani, supra note 31, at 423.
134. Id.
135. Which indeed is the case in many situations. Furthermore, if the manufacturer does not sell his software program to a distributor, there may exist some kind of licensing agreement such that the software manufacturer retains all liability for defects in the program which the quality control group did not, and could not, discover. Interview with Dr. H.
distributor as well, or employed a distributor but retained liability, then there would not be a need for strict liability because there would be privity between the buyer and the manufacturer.

It may be true that the injured person will often be the innocent party who has little or no knowledge of the computer software. A practical solution for the victim to recover would be for the injured person to sue the party who used the program and let that party implead the manufacturer. This would force the two parties to be on “equal footing,” with equal knowledge, to battle it out. For example, in a situation where a medical patient is injured by computer software that failed, the patient would sue the hospital (who would have some knowledge as to how the computer software program functioned), who in turn would bring in the manufacturer of the defective computer program. In this scenario, the injured party would be fully compensated.

C. Chilling Effect Upon Computer Software Manufacturers Will be Devastating

Finally, and most importantly, imposing strict products liability upon computer software manufacturers would undoubtedly prevent innovation in the computer software arena. Implementation of strict products liability would discourage the research and development of potentially life saving computer software. If computer software manufacturers were to face strict liability for defects in computer software which they did not know about and could not prevent, many manufacturers would discontinue “cutting edge” ventures, since one defect in the computer software may plunge the manufacturing entity into bankruptcy.

Additionally, imposing strict products liability upon computer software manufacturers for software defects which they did not know about would, in effect, make computer software manufacturers the insurer of their product. Strict liability, however, was never intended to make the manufacturer of a product the insurer of that product.136 Strict products liability would have more than a “chilling” effect upon computer software manufacturers, it would have a “freezing” effect.

Alkhatib, Professor of Computer Engineering, Santa Clara University, in Santa Clara, CA (Nov. 1, 1990).

VI. Conclusion

It is relatively clear that California courts could, under both Section 402A of the Restatement (Second) of Torts and under Barker v. Lull Engineering Co., apply strict products liability to computer software manufacturers if they so desired. The question is whether the policy considerations are strong enough for the courts to justify the progression from computer malpractice to liability without fault.

Based on the policy reasons considered in this Comment, this author believes that the courts have been justified and correct in not applying strict products liability to computer software manufacturers. In a modern society such as ours, where technology rapidly advances and changes and computer software is being used in more innovative situations, the "chilling" effect on technology as a result of imposing strict liability would be too great. Furthermore, the protection afforded under a warranty theory, negligence theory, and computer malpractice theory can adequately protect and compensate the consumer without preventing the innovation that is so desirable in our complex and technologically growing society. If strict liability were imposed upon computer software manufacturers, society could lose out on very important and potentially life saving computer software.