

ARTICLE**SOFTWARE PRODUCT LIABILITY: UNDERSTANDING AND MINIMIZING THE RISKS**

By *Lawrence B. Levy*[†]
and *Suzanne Y. Bell*^{††}

table of Contents

I. The Problem of Vendor Liability	1
II. Theories of Liability	2
A. Threshold Issue: Is Software a "Good" or a "Service"?	2
B. Contract Theories of Liability	7
C. Tort Theories of Liability	8
III. Minimizing the Risks	15
A. Contractual Provisions	15
B. Development and Marketing Strategies	23
C. Relationship With Licensees	26
D. Product Liability Insurance	27
IV. Conclusion	27

I. The Problem of Vendor Liability

As computer technology evolves, more powerful computer systems and software are available to more individual users and businesses.¹ As a result, software vendors are likely to face increasing exposure to lawsuits alleging that software did not perform as expected. The consequences of such lawsuits to software vendors could be catastrophic.

Several examples illustrate the extent of the potential liability faced by software vendors. In one case, an error ("bug") in a computerized therapeutic radiation machine caused it to administer incorrect dosages. Two people were killed and several others were seriously injured.² In another case, a construction company alleged that a bug in a spreadsheet program caused the company to underbid a \$3 million contract. The company sued the manufacturer of the program for \$245,000, claiming it had lost that amount as a result of the incorrect bid.³ Finally, in *Scott v. White Trucks*,⁴ the defendant's truck was equipped with computer-controlled anti-lock brakes. After the brakes failed and the truck crashed, the driver of the truck brought a product liability action alleging defects in the software.⁵

Few software product liability cases have been litigated, so little directly applicable judicial guidance is available to pinpoint steps a

software vendor should take to minimize liability. However, by understanding the legal theories upon which a hypothetical plaintiff may rely in a suit against a software vendor, it is possible to identify and understand where the risk of liability lies. This is the focus of Part II of this article. Part III will describe and provide examples of actions the vendor may take to minimize, shift, or spread the risk of liability for imperfect software. Such measures help protect the vendor and enable both the vendor and its customers to understand their respective rights and obligations. These measures also help to build a positive working relationship between the vendor and its customers by ensuring that expectations are realistic and by increasing the likelihood that potential problems with the software will be detected and corrected before they cause any loss or damage.

II. Theories of Liability

A. Threshold Issue: Is Software a "Good" or a "Service"?

A threshold issue is whether computer software is a "good" or a "service." One reason this is important is that sales of goods, but not of services, are subject to the damages and warranty provisions of the Uniform Commercial Code (UCC).⁶ Another reason is that manufacturers of products, but not providers of services, may be subject to suit under a strict liability theory of tort.⁷ A review of relevant cases shows that courts have used various analytical approaches to decide this issue.⁸

1. For Purposes of Applying the UCC

The goods vs. services question typically arises when courts must decide whether to apply the UCC. A few major variables have been important to courts in deciding this issue. For example, a court may consider whether the vendor included the software in a sale of hardware. If so, the software will be more likely to be considered a good than if the software had been sold by itself.⁹ A court may also ask whether the software was custom-designed or mass-marketed; the sale of the former has a greater likelihood of being characterized as a sale of services, while the latter is more likely to be considered a sale of goods.¹⁰ This test is more difficult to apply because often standard programs are altered to meet an individual customer's specific requirements, blurring the distinction between custom-designed and mass-marketed.

The court in *Data Processing Services, Inc. v. L.H. Smith Oil Corp.*¹¹ held that the sale of the software involved in the suit was a contract for services, not goods, because of both the custom nature of the software and the fact that there was no simultaneous sale of hardware. The court found:

The transaction here is clear-cut. Unlike many of the cases reported in other jurisdictions, DPS sold no "hardware" to Smith. Instead, DPS was retained to design, develop and implement an electronic data processing system to meet Smith's specific needs. ... We intentionally stress the active nature of DPS's role. The very terminology used by the trial court and the parties here show services, not goods were that for which Smith contracted.¹²

It is important to note that, although the court did not explain fully what it meant by its reference to the "terminology used by the trial court and the parties," it emphasized that the vendor's computer programmer had an active role in the development of the program for the specific customer.¹³

A jury apparently found another factor to be significant in *West Outer Drive Medical Center v. Compucare Inc.*¹⁴ In that case, a hospital contracted for the development of an "information system" consisting of both hardware and software.¹⁵ At trial, the software developer claimed that the hospital must have considered the software to be a good because it wanted to take advantage of investment tax credits that were available only if the software was tangible personal property.¹⁶ The jury found that the contract was for the sale of goods and that the UCC applied.¹⁷

Some courts have taken a different approach when a vendor provides not only the software itself, but also significant maintenance and support services. In such a transaction, a court must decide whether the goods or services aspect predominated in the transaction.¹⁸ If the services aspect predominated, the court will be more likely to characterize the transaction as a sale of services.¹⁹ In contrast, if the goods aspect predominated, the court will call the transaction a sale of goods.

Courts may decide whether the goods or services aspect of a transaction predominated by examining its financial details. In *Micro-Managers, Inc. v. Gregory*,²⁰ the court used this method of analysis to characterize a hybrid transaction as a sale of services. In that case, a petroleum corporation sued because the custom software it had purchased malfunctioned. The court noted that the corporation had paid \$59,828 to the software producer. Of this sum, \$55,968 had been paid for labor and support for the corporation's use of the software.²¹ The court held that this showed that the contract was primarily for services, and, consequently, that the UCC did not

apply.²² A court reached the opposite result using this method in *Stenograph Corp. v. MicroCAT Corp.*²³ It held that the UCC applied to the sale of all the assets and liabilities of a corporation because the parties had allocated \$950,000 of the total \$1,000,000 purchase price to the corporation's software, and only \$50,000 to the corporation's copyrights, trademarks, and other intangible assets.²⁴

Other courts take a less rigorous approach in analyzing hybrid transactions. The court in *RRX Industries, Inc. v. Lab-con, Inc.*²⁵ began by noting that it would search for "the essence of the agreement" using a "case-by-case analysis" because "software packages vary depending on the needs of the individual consumer."²⁶ In this case, the court continued, the transaction was a sale of goods, not services, because the sales aspect predominated.²⁷ This was true even though the software producer had provided a number of services, including employee training, repair services, and system upgrading. The court found that these services were only "incidental" to the sale of the software package.²⁸

2. For Purposes of Applying Strict Liability

Courts have not addressed the issue of whether software is a product for the purposes of applying strict liability in the same way that they have addressed the issue of whether software is a good for the purposes of applying the UCC. Certain cases, however, may be relevant.

In one series of cases, courts have held that information is a product and that strict liability therefore applies to it. First, in *Saloomey v. Jeppesen & Co.*,²⁹ navigational charts were found to be a product, not a service. In that case, inaccuracies in the charts caused a fatal airplane crash, and the decedents' administrators brought a wrongful death suit against the publisher of the charts.³⁰ The court said that since the charts were mass-produced and it was likely that purchasers substantially relied on them without making any alteration to them, the publisher had a duty to insure that consumers would not be injured by the use of the charts.³¹

Second, in *Brocklesby v. United States*,³² the court held a publisher of an instrument approach procedure for aircraft strictly liable for injuries incurred due to the faulty information contained in the procedure. Strict liability applied because the product was defective, even though the publisher had obtained the information from the government.³³

In contrast, other cases have not applied strict liability to information contained in books. For example, in *Cardozo v. True*,³⁴ a woman was poisoned when she ate an exotic ingredient called for by a recipe in a cookbook. The court held that the information contained in the book was not a product for the purposes of applying strict liability, and, therefore, the seller of the book was not liable on that theory for failure to warn of the nature of the ingredients used in the recipe.³⁵ These cases may be relevant because software usually either contains information or assists in the generation or manipulation of information.

There are strong policy reasons to apply the UCC to sales of software. For example, software sales would then be subject to a concise, statutory, and uniform body of law predictable to both vendors and users.³⁶ It would also be consistent with the parties' expectations; although they typically are licensees, especially when the same program is made available to multiple users, software users frequently regard themselves as purchasers of goods.³⁷ Finally, since the UCC provides for such measures as warranties, disclaimers, and limitations of liability,³⁸ the UCC provides a mechanism to allow the parties to allocate the risk of product performance.³⁹ However, if software is considered a "good" in order to apply the UCC, it increases the likelihood that it will be characterized as a good for other purposes, such as the application of strict product liability law.

Software vendors may be held liable for damages caused by their products under various contract and tort theories. This will be the focus of the remainder of Part II.

B. Contract Theories of Liability

Often a contract (such as a development or license agreement) exists between the injured user and the software vendor. If so, it will usually be the first place to which the user will turn for a theory of liability under which it may recover against the vendor for damages resulting from defects in the software. For example, the user may claim that the defect breached the terms of a software warranty contained in the contract. The elements and defenses of such a case would follow the normal rules of local contract law. In addition, if the transaction is determined to be a sale of goods,⁴⁰ the provisions of UCC Article 2 will apply to the interpretation of the contract and the recoverable damages.

Moreover, a court may hold a vendor liable under a warranty theory for statements about the software made outside the formal agreement.⁴¹ In addition, if a court determines that the transaction was a sale of goods subject to the UCC, in the absence of a warranty disclaimer it may hold the vendor liable for breach of warranties implied by that statute. One such warranty is the implied warranty of

merchantability, which requires that the software be reasonably fit for the general purpose for which it is sold.⁴² Another warranty is the implied warranty of fitness for a particular purpose, which applies when (1) the vendor knew of a particular purpose for which the software was required; and (2) the vendor knew that the user relied on the vendor's skill and judgment to furnish a suitable program.⁴³

Under a contract theory, a plaintiff may recover the difference between the market value of the software as delivered and the contract price of the software.⁴⁴ Plaintiffs may also be able to claim additional damages such as consequential damages,⁴⁵ incidental damages,⁴⁶ lost profits,⁴⁷ or other losses that the sellers should have reasonably anticipated, such as the loss to the construction company in completing the contract mentioned above.⁴⁸ The possibility of large damages from these additional categories poses a great threat to software vendors faced with a breach of contract claim.

Since these are contractual claims, the vendor frequently can draft its contracts to substantially limit its exposure.⁴⁹ However, as discussed below,⁵⁰ there may be limits to this ability to minimize liability.

C. Tort Theories of Liability

Since it may be possible for vendors to draft agreements that limit their exposure to contract damages for defective software,⁵¹ injured users also look to a variety of tort theories for recovery. Contractual liability disclaimers may not be effective in limiting vendor liability to suits brought under tort theories. In addition, injured third parties may rely on tort theories, whose applicability is not predicated upon the existence of a contractual relationship with the vendor.

1. Misrepresentation

One tort theory involves claims that the vendor fraudulently misrepresented the capabilities of the software.⁵² In order to prevail under this theory, the plaintiff must show that it was damaged because (1) the vendor misrepresented a material fact concerning the software, and (2) the plaintiff justifiably relied on this misrepresentation.⁵³ For example, in *Laurie Financial Corp. v. Burroughs Corp.*,⁵⁴ a vendor claimed that its computer system, including software, would be suitable for a prospective customer's business needs, when in fact the system was not.⁵⁵ The court held the vendor liable to the customer on a fraudulent misrepresentation theory. It found that the customer justifiably relied on the vendor's misrepresentations because the customer had no previous experience with the particular system involved and had not made an independent investigation of the system's capabilities.⁵⁶

The defendant's misrepresentation usually must be intentional for the plaintiff to recover,⁵⁷ but some jurisdictions recognize a cause of action for negligent misrepresentation.⁵⁸ Courts may allow some "puffing" in marketing efforts, but seem to be more inclined to find misrepresentation when the vendor makes misstatements concerning facts that are exclusively within the vendor's knowledge.⁵⁹

A fraudulent misrepresentation claim is especially threatening to software vendors because under this theory, a plaintiff may sue when it suffers damages solely to its intangible economic interests (such as business reputation), rather than personal injuries or damage to tangible personal property.⁶⁰

2. Negligence

A second tort theory is that the vendor was negligent in developing the software. The plaintiff must show that the vendor had a duty to use a specific standard of care, usually "reasonableness," and that the vendor breached that duty. There are many actions or omissions that might constitute such a breach. These might include, for example, a failure to (1) write or test the program properly, (2) correct significant bugs in the program, (3) warn of limitations in the program, (4) instruct users how to operate the program,⁶¹ or (5) provide adequate security for the system. The standard of care in each instance will depend on the specific circumstances. In addition, as technology evolves, there is the possibility that courts will hold vendors liable for less obvious breaches of duty such as the failure to insure that the software does not contain any hidden destructive programs (known as viruses).⁶² Finally, the plaintiff must show that the vendor's breach caused the plaintiff's injury, and that the plaintiff suffered damages from its injury.⁶³ Recoverable damages usually are restricted to bodily injury or property loss.⁶⁴

For a variety of reasons, plaintiffs have had less success claiming damages from software vendors under this theory than under other theories.⁶⁵ Demonstrating that a vendor's conduct is unreasonable is difficult and expensive. Moreover, the defenses of assumption of risk and contributory negligence are available to vendors. For example, vendors have successfully defended on the grounds that the buyer was negligent in using defective data or in hiring incompetent operators. As a result, it may be difficult to prove that the plaintiff's injury was caused by the vendor's breach of duty, such as supplying defective hardware or software, rather than by the

plaintiff's own error, such as use of incorrect data. Finally, some courts hold that a negligence action for economic loss is barred if a contract exists between the parties.⁶⁶

3. Professional Malpractice

A third tort theory is professional malpractice. In this variation on the negligence action,⁶⁷ the software vendor is characterized as a professional and therefore is held to owe to the plaintiff not merely a duty to act reasonably, but a higher duty to use a professional standard of care, analogous to the duty required of a physician or lawyer. This theory could apply only if the provision of software is characterized as a service, rather than as a sale of goods.⁶⁸

To date, courts have been reluctant to entertain a professional malpractice action against software vendors, principally because, unlike medicine and law, there are no established educational standards or regulations governing the performance of software programmers and developers, and they are not licensed as professionals.⁶⁹ However, an analysis of recent judicial and administrative decisions demonstrates that there appears to be some support in the courts for the imposition of a professional standard. In one case, a federal court suggested that persons engaging in computer consulting as part of a regulated profession may be held to the standard of care of that profession. In *Diversified Graphics, Ltd. v. Groves*,⁷⁰ a company hired a large accounting firm, Ernst & Whinney, to assist it in locating a turnkey computer system. When the system proved difficult to operate and failed to meet the company's data processing needs, it brought a negligence action against Ernst & Whinney. The court ruled that the firm should be held to the standard of care established by the rules for professional accountants. Specifically, it said that the American Institute of Certified Public Accountants' Management Advisory Services Practice Standards, which Ernst & Whinney had adopted for internal use, were applicable to the firm's provision of computer services.⁷¹ In addition, since there were no industry guidelines as to what is required of a turnkey system, the court detailed requirements of such a system.⁷² Although the case did not expressly find a cause of action for computer malpractice, in this instance it practically adopted one by employing a professional standard of care in a negligence action involving a defective computer system.

A court in an earlier case also provided support for holding computer programmers to a professional standard of care. In *Data Processing Services, Inc. v. L.H. Smith Oil Corp.*,⁷³ a case involving alleged failures of a computer programmer in designing data processing and accounting software for a customer, the Indiana Court of Appeals stated rather strong dictum that identified computer programmers with members of a profession:

Those who hold themselves out to the world as possessing skill and qualifications in their respective trades or professions impliedly represent they possess the skill and will exhibit the diligence ordinarily possessed by well informed members of the trade or profession.⁷⁴

The court also held that:

[t]he situation here is more analogous to a client seeking a lawyer's advice or a patient seeking medical treatment for a particular ailment than it is to a customer buying seed corn, soap, or cam shafts.⁷⁵

Finally, an administrative ruling held a computer programmer to the standard of care not of a professional computer programmer, but of the underlying profession that the program's functions performed. In Internal Revenue Service Revenue Ruling 85-189,⁷⁶ a programmer developed a program to prepare a user's personal income taxes. The program produced an incorrect return because the programmer failed to update it to reflect certain changes in the 1983 tax law. The IRS decided that the programmer would be held liable as a tax preparer for the deficiency in the user's return. In reaching this result, the IRS relied on Sections 6694(a) and 6694(b) of the Tax Code, which provide that a tax preparer may be subject to a penalty for any understatement due to either negligence or intentional disregard of the rules and regulations of the IRS.⁷⁷ It also relied on a previous ruling⁷⁸ that enunciates several factors that are taken into account in determining whether a preparer should be subject to a penalty: (1) the nature of the error that caused the understatement, (2) the frequency of the errors, (3) the materiality of the errors, and (4) the preparer's office practice.⁷⁹

Holding software programmers to a professional standard of care has advantages and disadvantages. On the one hand, as software programs become increasingly large and complex, it may be to the benefit of users that the programmers meet a particular standard of care in developing and debugging the program. On the other hand, to ascertain what standard of care should be applied would be extremely complex. Software programmers are diverse in their educational backgrounds,⁸⁰ and programming theories and techniques often are characterized as more of an art than a science. There are no established standards for computer professionals; standards would have to evolve on a case-by-case basis. Courts also would have to decide whether the standard varied by locale, as has been true for physicians,⁸¹ and whether there are specialties within the computer field for which the standard of care might differ.

The IRS's approach of applying the standard of care of the profession normally performing the function of the program could be applied to limit these problems. Consider again the situation of the people who were injured as a result of the malfunctioning of a program that regulated the amount of radiation exposure for patients undergoing cancer therapy.⁸² Under the IRS's approach,⁸³ a court would begin by finding that the administration of radiation would normally be the type of work done by a doctor. It would then apply the standard of care for doctors to the vendor's conduct, to determine whether the vendor breached a duty owed to the plaintiff. The court would therefore not have to find and apply a standard of care for the computer programmer profession.

However, there would also be drawbacks to this approach. The person writing software that, for example, generates tax returns or administers radiation, in most cases is probably not a professional tax preparer or physician, but merely relies on information given to him or her. It may not be fair to apply the standards of the underlying profession to the programmer, and if the law evolves in that direction, programmers may become reluctant to undertake such projects. There are also many professional tasks that could be performed by computer programs, necessitating provision of multiple standards of care applied in computer malpractice cases. In addition, not all computer programs perform the functions of a profession, and some may perform the functions of more than one profession. In those cases, courts would have to decide whether to apply a general reasonableness standard or the standard of a computer professional.

4. Strict Liability

Another tort theory that might be applicable is strict liability. It is the theory often used in cases involving defective consumer goods, and it would only apply if software is characterized as a product.⁸⁴ For strict liability to apply to the manufacturer of software, the user must have used the product in a reasonable fashion and the product must have reached the user without substantial change.⁸⁵ If the user is injured while using the product, the user need show only that the product caused the injury,⁸⁶ and that the product was sold in a defective or unreasonably dangerous condition.⁸⁷ The alleged defect could be a defect in the design or manufacturing of the software, or it could simply be a failure to warn of hazards.⁸⁸

An important feature of the strict liability theory is that it renders legally irrelevant the issue of whether the vendor acted reasonably.⁸⁹ By preventing the vendor from presenting exculpatory arguments, this theory in effect forces software manufacturers to guarantee the safety of their products.⁹⁰ The strict liability theory also has an effect on potential defendants and on recoverable damages. If it is applied, everyone in the chain of distribution of the product may be liable for the plaintiff's damages.⁹¹ However, users are not generally compensated for economic loss under a strict liability theory, but only for personal injury or property damage.⁹²

To date, courts have been reluctant to apply the theory of strict product liability to computer software.⁹³ Proponents have offered several arguments in favor of doing so. They argue that the software manufacturer should be held liable because it is in the best position to prevent defects, and can spread the risk of liability through insurance or by increasing the cost of the product; that strict liability assures compensation of the victim; that negligence is too difficult to prove; and that the application of strict liability will deter the manufacturer from producing defective software.⁹⁴ It is partially these types of considerations that have led courts to apply strict product liability to certain types of information.⁹⁵

On the other hand, the application of the strict liability theory might hinder the development of software. Consider again the medical program that regulates the amount of radiation exposure for cancer patients.⁹⁶ It is likely that clinical judgments and heuristic rules for making decisions, rather than fixed engineering and mathematical principles, would be used to develop the software.⁹⁷ The software developer is therefore not necessarily able to eliminate defects, any more than a medical practitioner can guarantee positive results from radiation treatments. Nevertheless, under the strict liability theory, the software manufacturer might be held liable for large personal injury damages if the radiation treatments do not help, or injure, the patient. Moreover, everyone in the chain of distribution may be liable, including the hospitals and medical practitioners that used the program.⁹⁸

In the coming years, we can expect to see each of these theories tested in the courts as the number of damage claims alleging defective software increases. The prudent software vendor should be cognizant of these risks and take actions to limit its exposure. We now turn to these preventive actions.

III. Minimizing the Risks

In Part II of this article, we outlined the legal theories upon which plaintiffs may rely in actions against software vendors resulting from software not performing as expected. In this Part, we focus on measures a software vendor may take to minimize, shift, or spread the risk of liability posed by these theories.

A. Contractual Provisions

Software vendors can limit their liability for defective performance of their products through a variety of contractual provisions. There are some general limits to the effectiveness of these measures. For mass-distributed software that is packaged with a shrink wrap license,⁹⁹ the enforceability of an exculpatory agreement is questionable, especially if it excessively favors the vendor.¹⁰⁰ For custom software development and license agreements, customers often resist contractual limitations upon liability and demand that the developer assume the risk of the software failing to meet agreed-upon specifications. The developer will then have to factor such risk into the price of the software. Moreover, contractual provisions are less effective, if at all, in limiting the vendor's liability to injured third parties.¹⁰¹ Despite these limitations, such contractual provisions can effectively reduce the vendor's exposure.

1. Warranty Disclaimers

A software vendor may disclaim all warranties with respect to the performance of the software, or all warranties except those expressly stated in the contract. For example, in *Meeting Makers, Inc. v. American Airlines, Inc.*,¹⁰² a court dismissed with prejudice claims brought by a purchaser of a computer system because "[t]he contracts contained conspicuous disclaimers of all warranties of merchantability or fitness for a particular purpose."¹⁰³ A sample warranty disclaimer is provided below.¹⁰⁴

Courts generally uphold warranty disclaimers except in cases of "unconscionability," which means that the disclaimer provision is excessively one-sided, oppressive, or surprising to the purchaser.¹⁰⁵ This limit may be particularly relevant in the case of mass-distributed software accompanied by shrink-wrap licenses. It is possible that a court would characterize the license included in the package as a contract of adhesion, and find the broad disclaimers contained within the license to be unconscionable and unenforceable.¹⁰⁶ For this reason, it is often advisable to include in a shrink-wrap license a limited form of express warranty.¹⁰⁷ In addition, it may be difficult for a vendor to negotiate a disclaimer in contracts for custom software.

Warranty disclaimers also should disclaim any warranties that may be implied by law. If a court determines that the vendor's software is a product and not a service, the UCC will apply to its sale¹⁰⁸ and the software will be covered by the UCC's implied warranties of merchantability and fitness for a particular purpose.¹⁰⁹

Courts uphold disclaimers of implied warranties as long as such warranties use the exact words required by the UCC and are conspicuous.¹¹⁰ A disclaimer is conspicuous when it is "so written that a reasonable person against whom it is to operate ought to have noticed."¹¹¹ This is a fact-based inquiry performed by the court.¹¹² In *Sierra Diesel Injection Serv., Inc. v. Burroughs Corp.*,¹¹³ the court held that:

Whether a disclaimer is conspicuous is not simply a matter of measuring the type size or looking at the placement of the disclaimer within the contract. A reviewing court must ascertain that a reasonable person in the buyer's position would not have been surprised to find the warranty disclaimer in the contract.¹¹⁴

In *Hunter v. Texas Instruments, Inc.*,¹¹⁵ the court found the disclaimer conspicuous because it, and the statement directing the purchaser to read the disclaimer, were in larger print than the surrounding text.¹¹⁶

There is another limit to the use of disclaimers of implied warranties. Under the Magnuson-Moss Warranty-Federal Trade Commission Improvement Act,¹¹⁷ if a consumer good is covered by a written warranty, the vendor cannot disclaim implied warranties arising under state law.¹¹⁸ The vendor may, however, limit the duration of the implied warranties to that of the written warranty.¹¹⁹

A vendor may want to consider providing a limited warranty rather than disclaiming all warranties. This may be preferable from a marketing standpoint and may decrease the likelihood that a contract will be found unconscionable. Moreover, it may be easier to obtain a limited warranty than a disclaimer of all warranties in negotiations with purchasers of custom software. A limited warranty may provide, for example, that the software will perform according to certain specifications for a limited period of time. Alternatively, the vendor simply may specify the level of support and bug fixing it will provide.¹²⁰ In either case, the vendor commits itself to the specified warranty or service, but does not extend as complete a warranty as might have been requested by the purchaser, or implied by the UCC.

2. Limitation of Remedies

A software vendor may effectively limit its exposure to monetary damages by providing in the contract that, in the event of a breach,

the purchaser's remedies are limited to repair or replacement of the defective software, or to the price paid for the software.¹²¹ For example, in *Office Supply Co. v. Basic/Four Corp.*,¹²² the plaintiff sought \$186,000 in compensation for "lost customers, income, good will and executive time ... additional hardware and software expense, ... personnel expense and maintenance expense" allegedly caused by the vendor's defective hardware and software.¹²³ The court noted that the parties' contract contained a provision limiting the plaintiff's remedy to repair or replacement of the defective parts.¹²⁴ As a result, it granted summary judgment against the plaintiff with respect to this claim.¹²⁵

A limitation of remedies provision may not be upheld when the proffered remedy "fail[s] of its essential purpose."¹²⁶ This may be the case, for example, when the software does not perform as warranted and the vendor is unable to repair or replace the product.¹²⁷ If a remedy fails of its essential purpose, a buyer may generally pursue the full range of contract remedies.¹²⁸ Thus in *RRX Industries, Inc. v. Lab-con, Inc.*,¹²⁹ the trial court found that the software developer was "either unwilling or unable to provide a system that worked as represented, or to fix the 'bugs' in the software..."¹³⁰ On appeal, the Ninth Circuit agreed that "the default of the seller [was] so total and fundamental that its consequential damages limitation was expunged from the contract."¹³¹

The following is a sample warranty disclaimer that includes a limitation of remedies provision:

Vendor warrants that, for a period of ninety (90) days from the date of delivery to you the diskettes on which the program is furnished under normal use will be free from defects in materials and workmanship and the program under normal use will perform substantially in accordance with the specifications contained in [specified document]. Vendor's entire liability and your exclusive remedy under this warranty will be, at Vendor's option, to use reasonable commercial efforts to attempt to correct or work around errors, to replace the program or diskettes with functionally equivalent software or diskettes, as applicable, or to refund the purchase price and terminate this Agreement.

EXCEPT FOR THE ABOVE EXPRESS LIMITED WARRANTIES, VENDOR MAKES AND YOU RECEIVE NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY OR IN ANY COMMUNICATION WITH YOU, AND VENDOR SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Vendor does not warrant that the operation of the program will be uninterrupted or error free.¹³²

3. Limitation of Liability

A software vendor may contract for limitations on its liability if the software later proves to be defective. One way to do this is to limit liability for certain types of damages.¹³³ For example, a vendor may state that it will not be held liable for any special, consequential, or incidental damages, or for the cost of procurement of substitute goods or services. In *Harper Tax Services, Inc. v. Quick Tax Ltd.*,¹³⁴ a customer was barred from seeking consequential damages on its breach of contract claim against the software vendor, Quick-Tax, because their agreement specified that:

Quick-Tax will not be liable for any lost profits, or for any claims against the Customer by any other party, except a claim for patent or copyright infringement as provided herein, [and] IN NO EVENT WILL QUICK-TAX BE LIABLE FOR CONSEQUENTIAL DAMAGES EVEN IF QUICK-TAX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.¹³⁵

Another method is to specify liquidated damages. The amount of damages must be:

... reasonable in the light of the anticipated or actual harm caused by the breach, the difficulties of proof of loss, and the inconvenience or nonfeasibility of otherwise obtaining an adequate remedy.¹³⁶

However, an "unreasonably large" liquidated damages term is void as a penalty.¹³⁷

As with the other contractual provisions discussed in this section, limitations on liability generally are upheld in the commercial setting,¹³⁸ although they are narrowly construed. As a result, care must be taken to incorporate them in a contract in a manner that promises the broadest application and that limits the likelihood that the limitation will be held not to apply to all provisions of the contract.¹³⁹ In addition, a plaintiff may assert the doctrine of unconscionability against the enforcement of these provisions.¹⁴⁰

The following is a sample limitation of liability clause:

IN NO EVENT WILL VENDOR BE LIABLE FOR ANY DAMAGES, INCLUDING LOSS OF DATA, LOST PROFITS, COST OF COVER OR OTHER SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES ARISING IN ANY WAY OUT OF THIS AGREEMENT, OR FROM THE USE OF THE PROGRAM OR ACCOMPANYING DOCUMENTATION, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY. THIS LIMITATION WILL APPLY EVEN IF VENDOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND NOTWITHSTANDING THE FAILURE OF ANY LIMITED REMEDY PROVIDED HEREIN. CUSTOMER ACKNOWLEDGES THAT THE LICENSE FEE REFLECTS THIS ALLOCATION OF RISK.

4. Integration Clause

A software vendor may use an integration clause to attempt to protect itself against liability for statements made outside the contract concerning the performance of the software. Such a clause states that the entire agreement between the parties is contained in their contract. For example, in *Kalil Bottling Corp. v Burroughs Corp.*,¹⁴¹ a soft drink bottler purchased a computer system to improve its inventory and accounting operations. After various problems, the bottler purchased a new computer from a different manufacturer and sued the seller of the first system on various theories of liability. On appeal from a verdict for the buyer, the Arizona Court of Appeals ruled that an integration clause in the parties' agreement negated the plaintiff's claims of fraud, consumer fraud, and negligent misrepresentation through the operation of the parol evidence rule.¹⁴²

In general, integration clauses have been effective in voiding representations about the software made outside the contract.¹⁴³ However, this is not always the case. In *Sierra Diesel Injection Serv., Inc. v. Burroughs Corp.*,¹⁴⁴ the court held that the buyer of a computer system was entitled to rely on representations contained in a letter despite the disclaimer and integration clause contained in the contract. The court reasoned, in part, that the representations constituted express warranties which were part of the agreement between the parties; and that it was reasonable for the purchaser not to rely on only one document for the complete agreement of the parties, but on the series of contracts that were part of the transaction.¹⁴⁵ In addition, integration clauses will not be upheld when fraudulent statements of the vendor caused the purchaser to enter into the contract. In such a case, the fraud invalidates the entire contract.¹⁴⁶

The sample warranty disclaimer presented above¹⁴⁷ attempts to prevent prior communications from becoming contractual warranties by its disclaimer of any warranties "in any communication with you." The following is an example of a standard integration clause:

THIS AGREEMENT SETS FORTH THE ENTIRE AGREEMENT AND UNDERSTANDING OF THE PARTIES RELATING TO THE SUBJECT MATTER HEREOF AND SUPERSEDES ALL PRIOR AGREEMENTS, DISCUSSIONS AND UNDERSTANDINGS BETWEEN THEM, WHETHER ORAL OR WRITTEN, RELATING TO THE SUBJECT MATTER HEREOF.

5. Parties' Duties and Standard of Care

In tort litigation, the vendor will be held to a vague standard of reasonableness¹⁴⁸ (assuming that a professional standard of care is not applied), unless the parties provide otherwise in their contract. A contractual standard of care can attempt to define what is "reasonable" and provide a more concrete guide by which the parties may judge the vendor's conduct. For example, the contract could specify the obligations of the vendor, such as the time within which it must fix bugs;¹⁴⁹ and the level of training to be provided.

The contract may also specify certain obligations on the part of the buyer. For example, it may impose obligations to notify the vendor of bugs within a certain time period after discovery, to meet standards for installation and facilities, to provide a certain level of training to end users, to create and keep current appropriate back up programs, and to provide suitable input data. In addition, the vendor may impose an obligation to hire operators having certain qualifications. For example, in the medical context, the contract may state that only certain licensed professionals should operate the software.

Having specified the parties' standard of care in the contract, the software vendor, if it is sued, may contend that it is not negligent because it met its contractual obligations. In addition, it may contend that it should not be held liable for the defective performance of the software if the buyer failed to live up to its obligations. Moreover, both parties may benefit from a clear statement of their respective duties. This may engender a better working relationship between the parties, increase the probability that the software is properly used and maintained and help to identify defects before they cause loss or damage.

6. Statute of Limitations

The UCC permits a software vendor to add to the contract a clause to shorten the time period within which a buyer may bring suit. The parties may reduce this period to not less than one year after the cause of action accrues.¹⁵⁰ Assuming that the UCC applies, the developer may want to include such a clause to avoid liability for claims that accrue long after the development work has been completed. However, a contractual provision specifying the statute of limitations applies only to contractual claims against the vendor, not to tort claims.¹⁵¹

B. Development and Marketing Strategies

Despite their advantages, contractual provisions may not suffice to protect the software vendor against liability for defective software, especially to third parties not in privity of contract with the vendor.¹⁵² However, there are a variety of product development and marketing measures that a vendor should consider that might limit its exposure to such liability. Unfortunately, some of these measures may affect the marketability of its product. Therefore, in deciding whether to adopt them, the vendor must weigh this disadvantage against its need to protect against possible liability. In assessing the latter risk, the vendor should realize that its exposure to liability varies with the type of program it is marketing. For example, such risk will be dramatically different for a word processing program than for an expert system in a manufacturing plant.

1. Product Development Strategies

Software vendors may take a variety of measures in designing their products to minimize exposure to liability. Again, the specific measures to be taken will depend on the type of software and its intended use. As a result, it is difficult to create a generic set of standards which should be applicable to all software development. The following are some ideas which may be appropriate in particular circumstances:¹⁵³

1. Use qualified software programmers.
2. Document each step in the process, to show who wrote the program and the care taken in each step.
3. Test the software adequately and thoroughly before its release.
4. It may be prudent to have a different group of employees test the software than the group that developed it.
5. Take actions to avoid errors in the substantive information incorporated into the program. The vendor should keep records of how such technical information is incorporated, and who provided it.¹⁵⁴ To the extent possible, the vendor should confirm the accuracy of this information through several sources.¹⁵⁵
6. If the software is being developed for a particular industry and there are applicable professional codes of conduct for that industry, it may be prudent to follow these codes.

By taking these steps, the vendor will not only keep the software as free of technical error as is possible; it will be able to document its level of care should the issue be raised in a lawsuit.

The vendor may also consider incorporating functional limitations into the software in order to minimize the possibility of serious error. Although a developer may be capable of designing a more sophisticated system, it may be prudent to include certain limitations in the system; for example, that the system merely recommends a certain action, but does not act or make a final decision. To return to the radiation machine example,¹⁵⁶ the case shows that it might be risky if the software controlling therapeutic machinery goes beyond recommending a "standard" dosage or course of action requiring human judgment and intervention to put into action. By including such limitations, the vendor reduces the risk that its program will be held responsible for an erroneous course of action that leads to a plaintiff's injury. However, the same limitations may make the program less attractive to users.

In the case of custom software programs, it is particularly important for the programmer to work closely with the customer while developing the program. As a result of ongoing communications, the programmer can create a program that meets as closely as possible the needs of the customer. At the same time, the programmer can create accurate expectations on the part of the customer regarding the capabilities of the system, and fully explain to the customer the limitations of the system.¹⁵⁷

2. Marketing Strategies

In addition to product development strategies, software vendors may use a variety of marketing measures to minimize their exposure to liability.

The most fundamental principle is that the vendor should be careful not to make false claims in marketing information or other sales communications. Under the UCC, express warranties can be created by words, written or oral, or by conduct; and a court could hold a vendor liable for statements outside the license agreement.¹⁵⁸ A court may also hold a vendor to its marketing statements on a fraudulent misrepresentation theory.¹⁵⁹ Therefore, it is wise to be cautious about making such statements, even if the vendor's license agreement contains an integration clause.¹⁶⁰

In addition, the vendor should consider warning users of limitations in the software. If appropriate, the warning also should specify that the program is not to be used in situations in which the failure to perform could result in bodily injury. The warning may be placed in the license agreement, on the product packaging, on the first screen of the program to appear to the user, or on any other place where it will be clearly visible to the user of the program.

Finally, the vendor should exercise appropriate guidance at every level of its distribution channels. This might include ensuring that distributors and dealers are properly trained and cannot change the software or, if they can, that they will be fully liable for damages incurred as a result of such changes. The vendor and its distributors also should be careful not to market the system without providing adequate documentation and instructions on proper use.

The vendor's vigilance might also extend to its customers. For example, the vendor might establish a means by which users can report bugs. If a significant bug is discovered, it may be advisable to promptly notify all users even if an update is not yet available. When updates and error corrections become available, it may be desirable to require the user to substitute them for the old program. To achieve these goals, the vendor may want to require users to be registered. More aggressive tactics include requiring the buyer to acknowledge the various warnings, disclaimers, and limitations in writing; imposing penalties for breaches of the restrictions on use; and auditing user compliance with the restrictions.¹⁶¹

The vendor may wish to preserve its ability to update and correct its products by reserving the right to withdraw the software from its customers. Of course, the extent to which any of these actions is called for depends on many factors, including the function of the software, the likelihood of damage, the sophistication of the users, and the vendor's risk profile.

C. Relationship With Licensees

As licensors of software, vendors continue to run some risk of liability for the acts of their licensees. For example, the vendor may still be liable to users of the program under the tort theories of liability discussed above,¹⁶² and a user may be able to assert vendor liability based on the vendor's negligent selection of a licensee or distributor. There are several steps a vendor can take to reduce the possibility of such liability.

First, the vendor should exercise care in selecting licensees that are technically competent and financially secure.¹⁶³ Of course, the degree of competence and security required depends on the type of software, the sophistication of the users, and other factors.

Second, the licensor should attempt to structure the relationship in the manner least likely to produce liability for the acts of its licensees. Viewed along a continuum from the most likely to least likely to be held liable, partners and joint venturers are liable for each other's actions within the scope of the partnership or the joint venture relationship; if the licensor controls and directs the work product of the licensee, it may be argued that the licensee was the employee or agent of the licensor, rendering the licensor liable for the actions of the licensee;¹⁶⁴ licensors may be held liable for the acts of their subsidiary-licensees under alter ego or fraud theories; however, independent licensors generally are not liable for the acts of their licensees.¹⁶⁵ (Of course, in each case, the licensor may be held liable for its own negligence, such as supplying a licensee with defective products or negligently selecting a licensee). The vendor should take appropriate steps to tailor its relationship with its licensee to reflect the degree of liability and control it desires.

Finally, the vendor should consider requesting from the licensee an agreement to indemnify the licensor for losses incurred as a result of the licensee's actions. Courts usually uphold such clauses, although they tend to construe them narrowly.¹⁶⁶

D. Product Liability Insurance

Finally, software vendors may lessen the impact of huge liability verdicts by purchasing product liability insurance. However, the cost and availability of this insurance can present problems.¹⁶⁷ In the case of expert systems, for which the developer may be held to the

IV. Conclusion

As society increasingly relies on software to perform critical functions in everything from manufacturing to life support systems, the risk that an error in a software program will lead to economic loss, property damage, or personal injury increases. Prudent software developers will be cognizant of these risks and will take steps to minimize their exposure to this type of liability. Determining how far a software developer should go in this effort requires balancing the degree of exposure to software product liability against the adverse impact, if any, of the steps required to limit this exposure on the software vendor's ability to market and sell its software.

†1990 Lawrence B. Levy and Suzanne Y. Bell

† Partner, Wilson, Sonsini, Goodrich & Rosati, Palo Alto, CA. Indiana University, B.S. (with honors) 1980; Harvard Law School, J.D. (*cum laude*) 1983.

†† Associate, Wilson, Sonsini, Goodrich & Rosati, Palo Alto, CA. Middlebury College, B.A. (*cum laude*) 1980; Columbia University, M.S. 1982; Stanford Law School, J.D. (with distinction) 1988.

The authors gratefully acknowledge the contribution of Paul Rothstein, a student at Stanford Law School, for his assistance in the preparation of this article.

1. Raysman & Brown, *Strict Product Liability for Software and Data*, N.Y.L.J., Sept. 15, 1988 at 3, 3; Gemignani, *Product Liability and Software*, 8 RUTGERS COMPUTER & TECH. L.J. 173, 173-75 (1981).

2. Zammit & Savio, *Tort Liability For High Risk Computer Software*, 23 PLI/PAT 373, 375 (1987).

3. Blodgett, *Suit Alleges Software Error*, A.B.A. J., Dec. 1, 1986, at 22.

4. 699 F.2d 714 (5th Cir. 1983).

5. *Id.* at 723.

6. *See infra* note 40 and accompanying text.

7. *See infra* note 84 and accompanying text.

8. Software is generally licensed and not sold, and the UCC by its terms applies to sales. However, when courts have found that software is a "good," they have not allowed the fact that software is licensed to insulate vendors from liability under the UCC.

9. *See* Comment, "*Computer Malpractice*" and Other Legal Problems Posed by Computer "Vaporware," 33 VILL. L. REV. 835, 855 (1988).

10. *See infra* notes 11-13 and accompanying text.

11. 492 N.E.2d 314 (Ind. Ct. App. 1986).

12. *Id.* at 318.

13. *Id.*

14. No. 80-73315, E.D. Mich., Southern Div., *discussed in Software Contract Held to be Sale of Goods*, 6 COMPUTER L. STRATEGIST 1, 8 (July 1989).

15. *Id.*
16. *Id.*
17. *Id.*
18. In these "hybrid" cases, courts consider the software to be a "good" and then proceed to the question of whether the goods or services aspect of the transaction predominated. *See* Triangle Underwriters, Inc. v. Honeywell, Inc., 457 F. Supp. 765, 769 (E.D.N.Y. 1978), *modified* 604 F.2d 737 (2d Cir. 1979) ("The system was subject to sale, and the services provided ... were incidental to that sale."); *see also* Chatlos Sys., Inc. v. Nat'l Cash Register Corp., 479 F. Supp. 738, 742-43 (D.N.J. 1979).
19. *See infra* notes 20-24 and accompanying text.
20. 147 Wis. 2d 500, 434 N.W.2d 97 (Ct. App. 1988).
21. *Id.* at 508, 434 N.W.2d at 100.
22. *Id.* at 509, 434 N.W.2d at 100.
23. 1987 U.S. Dist LEXIS 6442 (N.D. Ill. July 10, 1987) (motion to dismiss or strike plaintiff's claims), 1988 U.S. Dist LEXIS 2030 (N.D. Ill. Mar. 4, 1988) (motion to dismiss plaintiff's amended complaint), 1989 U.S. Dist LEXIS 10105 (N.D. Ill. Aug. 21, 1989) (motion to dismiss defendants' amended counterclaim).
24. 1987 U.S. Dist LEXIS 6442 at 12.
25. 772 F.2d 543 (9th Cir. 1985).
26. *Id.* at 546.
27. *Id.*
28. *Id.*; *see also* Chatlos Sys., Inc. v. Nat'l Cash Register Corp., 479 F. Supp. 738, 742-43 (D.N.J. 1979); Triangle Underwriters, Inc. v. Honeywell, Inc., 457 F. Supp. 765, 769 (E.D.N.Y. 1978), *modified* 604 F.2d 737 (2d Cir. 1979).
29. 707 F.2d 671 (2d Cir. 1983) (applying Colorado law).
30. *Id.* at 672-73.
31. *Id.* at 676-77.
32. 767 F.2d 1288 (9th Cir. 1985).
33. *Id.* at 1294-96.
34. 342 So. 2d 1053 (Fla. Dist. Ct. App. 1977).
35. *Id.* at 1056-57.
36. *See* Rodau, *Computer Software: Does Article 2 of the Uniform Commercial Code Apply?*, 35 EMORY L.J. 853, 857-60 (1986).
37. Birnbaum, *Strict Products Liability And Computer Software*, 8 COMPUTER/L. J. 135, 148 (1988); Comment, *Computer Software and Strict Products Liability*, 20 SAN DIEGO L. REV. 439, 453 (1983).

38. *See infra* notes 102-40 and accompanying text.

39. Zammit & Savio, *supra* note 2, at 382.

40. "Unless the context otherwise requires, this Article applies to transactions in goods...." U.C.C. § 2-102 (1989); *see supra* notes 6-39 and accompanying text.

41. Levy, *Software Warranties: Multiple Issues and Drafting Considerations*, 5 *COMPUTER LAW*. 14, 15-16 (1988).

42. "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." U.C.C. § 2-314(1) (1989). "Goods to be merchantable must be at least such as ... are fit for the ordinary purposes for which such goods are used" *Id.* § 2-314(2)(c).

43. *Id.* § 2-315.

44. *Id.* § 2-713(1).

45. *Id.* §§ 2-713(1), 2-715(1).

46. *Id.* §§ 2-713(1), 2-715(2).

47. *Id.* § 2-715(2).

48. *See supra* note 3 and accompanying text.

49. In Pennsylvania, privity of contract is not required to recover for breach of UCC warranties of merchantability and fitness for a particular purpose under Sections 2-314 and 2-318 of the Pennsylvania Uniform Commercial Code. Holding that Section 2-318 should be co-extensive with the state's doctrine of strict product liability, a Pennsylvania court allowed a purchaser to recover damages from a computer manufacturer for breach of implied warranty even though the purchaser had actually purchased the computer system from a reseller who had purchased it from the computer manufacturer. The disclaimers and limitations in the manufacturer's contract with the reseller did not apply to the reseller's customer. *Spagnol Enter., Inc. v. Digital Equipment Corp.*, 568 A.2d 948, 390 Pa. Super. 372 (1989).

50. *See infra* notes 99-101, 105-106 and accompanying text.

51. *See infra* notes 102-51 and accompanying text.

52. Plaintiffs apparently have had some success under this theory. Reece, *Liability for Defective Computer Software*, 1987 *COMPUTER L. REP.* 853, 855.

53. Kopf, *Debugging the Computer Contract: A Preventive Strategy Based on a Review of Vendor Liability*, 20 *U.C.C. L.J.* 130, 154 (1987).

54. No. 86 C 10231, published in 1985 *COMPUTER L. REP.* 630.

55. *Id.* at 631-32.

56. *Id.* at 634-35.

57. *See, e.g.*, *Harper Tax Servs., Inc. v. Quick Tax Ltd.*, 686 F. Supp. 109, 113 (D. Md. 1988) (material representation of fact must be "false and known false by the defendant").

58. See Harper Tax Servs., Inc., 686 F. Supp. at 114 ("Under New York law a claim of negligent misrepresentation is stated only in special circumstances and upon certain allegations including the existence of a nexus of intimacy between the parties approaching that of privity and closer than that of ordinary buyer and seller."); Conley, *Tort Theories of Recovery Against Vendors of Defective Software*, 13 RUTGERS COMPUTER & TECH. L.J. 1, 20-22 (1987).
59. See Computer Sys. Eng'g, Inc. v. Quantel Corp., 740 F.2d 59, 65-66 (1st Cir. 1984) (fraud claim upheld on appeal where purchaser neither knew nor could have known of defects in software).
60. Gregg & Folk, *Liability for Substantive Errors in Computer Software*, COMPUTER L. REP. 18, 20 (1986) (citing W. PROSSER & W. KEETON, PROSSER AND KEETON ON TORTS 657 (5th ed. 1984)).
61. See Midgely v. S.S. Kresge Co., 55 Cal. App. 3d 67, 127 Cal. Rptr. 217 (1976), discussed in Birnbaum, *supra* note 37, at 150-51.
62. For a catalog of such "rogue computer programs," see Branscomb, *Rogue Computer Programs and Computer Rogues: Tailoring the Punishment to Fit the Crime*, 16 RUTGERS COMPUTER & TECH. L.J. 1, 4-5 (1990).
63. Zammit & Savio, *supra* note 2, at 391.
64. *Id.* at 396. *Contra* Invacare Corp. v. Sperry Corp., 612 F. Supp 448, 454 (N.D. Ohio 1984).
65. Reece, *supra* note 52, at 855.
66. Office Supply Co. v. Basic/Four Corp., 538 F. Supp. 776, 791-92 (E.D. Wis. 1982); Reece, *supra* note 52, at 855; Kopf, *supra* note 53, at 149.
67. See *supra* notes 61-66 and accompanying text.
68. Zammit & Savio, *supra* note 2, at 397.
69. See, e.g., Chatlos Sys., Inc. v. Nat'l Cash Register Corp., 479 F. Supp. 738, 740 n.1 (D.N.J. 1979); Triangle Underwriters, Inc. v. Honeywell, Inc., 457 F. Supp. 765, 770-71 (E.D.N.Y. 1978), modified 604 F.2d 737 (2d Cir. 1979); Zammit & Savio, *supra* note 2, at 397-98.
70. 868 F.2d 293 (8th Cir. 1989).
71. *Id.* at 296-97.
72. *Id.* at 297.
73. 492 N.E.2d 314 (Ind. Ct. App. 1986).
74. *Id.* at 319.
75. *Id.*
76. Rev. Rul. 85-189, 1985-2 C.B. 341.
77. *Id.* at 342.
78. Rev. Proc. 80-40, 1980-2 C.B. 734.
79. Rev. Rul. 85-189, 1985-2 C.B. at 342.

80. Standards and trends in software programming education are discussed in STAFF OF SUBCOMM. ON INVESTIGATIONS AND OVERSIGHT, HOUSE COMM. ON SCIENCE, SPACE, AND TECHNOLOGY, 101ST CONG., 1ST SESS., BUGS IN THE PROGRAM: PROBLEMS IN FEDERAL GOVERNMENT COMPUTER SOFTWARE DEVELOPMENT AND REGULATION 23-25 (Comm. Print 1989).
81. See W. PROSSER, HANDBOOK OF THE LAW OF TORTS 164 (4th ed. 1971).
82. See *supra* note 2 and accompanying text.
83. See *supra* notes 76-79 and accompanying text.
84. "Thus far efforts to find strict liability as to the services themselves have entirely failed." *Id.* at 679, *quoted in* Birnbaum, *supra* note 37, at 146.
85. As stated by the court in *Scott v. White Trucks*, a plaintiff "without fault on his part" may recover under this theory if he "proves ... that the product was defective when it left the hands of the manufacturer." 699 F.2d 714, 716-17 (5th Cir. 1983), (quoting *Weber v. Fidelity & Casualty Ins. Co.*, 250 So. 2d 754, 755 (La. 1971), and *Madden v. Louisiana Power & Light Co.*, 33 So. 2d 249, 253 (La. Ct. App. 1976)).
86. *Scott*, 699 F.2d at 716 (quoting *Weber v. Fidelity & Casualty Ins. Co.*, 250 So. 2d at 755).
87. RESTATEMENT (SECOND) OF TORTS § 402A (1965).
88. Zammit & Savio, *supra* note 2, at 385-86.
89. *Id.* at 384-85.
90. *Id.*
91. RESTATEMENT (SECOND) OF TORTS, *supra* note 87, § 402A; Raysman & Brown, *supra* note 1, at 3.
92. Raysman & Brown, *supra* note 1, at 4; Frank, *Tort Adjudication and the Emergence of Artificial Intelligence Software*, 21 SUFFOLK U. L. REV. 623, 647 (1987). A Pennsylvania court, however, held that a purchaser may recover for a breach of statutory warranties against a remote manufacturer for purely economic loss. *Spagnol Enters., Inc. v. Digital Equipment Corp.*, 568 A.2d 948, ___ Pa. Super. ___ (1989), *discussed supra* note 49.
93. See Conley, *supra* note 58, at 29-30.
94. Note, *Developing a New Set of Liability Rules for a New Generation of Technology: Assessing Liability for Computer-Related Injuries in the Health Care Field*, 7 COMPUTER/L. J. 517, 530 (1987); Birnbaum, *supra* note 37, at 135.
95. See *supra* notes 29-33 and accompanying text.
96. See *supra* note 2 and accompanying text.
97. Lawrence, *Strict Liability, Computer Software and Medicine: Public Policy at the Crossroads*, 23 TORT & INS. L.J. 1, 15 (1987). This is different from, e.g., *Saloomey v. Jeppesen & Co.*, 707 F.2d 671 (2d Cir. 1983), in which the error was a navigational chart that incorrectly indicated that an airport had a full instrument landing system. *Id.* at 673.
98. Note, *supra* note 94, at 531.
99. This type of license accompanies the software but is not signed by the user. It gives the purchaser the right to return the software if the purchaser refuses to agree to the terms of the license. See Note, *The Enforceability of State "Shrink-Wrap" License Statutes in Light of Vault Corp. v. McQuaid Software, Ltd.*, 74 CORNELL L. REV. 222, 233 (1988).

100. Levy, *supra* note 41, at 17.

101. *See supra* notes 51-52 and accompanying text.

102. 513 So. 2d 700 (Fla. Dist. Ct. App. 1987).

103. *Id.* at 701.

104. *See infra* note 132 and accompanying text.

105. *See, e.g.,* Badger Bearing Co. v. Burroughs Corp., 444 F. Supp. 919, 923 (E.D. Wisc. 1977) (rejecting claim of unconscionability); Reece, *supra* note 52, at 854-55.

106. Sullivan, *Minimizing Exposure to Strict Products Liability as a Technology Licensor*, 9 Licensing L. & Bus. Rep. 73, 76 (1986); Vault Corp. v. McQuaid Software Ltd., 655 F. Supp. 750 (E.D. La. 1987), *aff'd* 847 F.2d 255 (5th Cir. 1988).

107. Levy, *supra* note 41, at 17.

108. *See supra* notes 6-39 and accompanying text.

109. U.C.C. §§ 2-314, 2-315 (1989).

110. "[T]o exclude or modify the implied warranty of merchantability or any part of it the language must mention merchantability and in case of a writing must be conspicuous, and to exclude or modify any implied warranty of fitness the exclusion must be by a writing and conspicuous. Language to exclude all implied warranties of fitness is sufficient if it states, for example, that 'There are no warranties which extend beyond the description on the face hereof.'" *Id.* § 2-316(2). *See also* Reece, *supra* note 52, at 855; Comment, *supra* note 9, at 867.

An implied warranty also can be excluded or modified by "course of dealing, course of performance, or trade usage." U.C.C. § 2-316 (3)(c). In addition, where the buyer examined the goods (or a sample or model of them) before entering into the sales contract, there are no implied warranties "with regard to defects which an examination ought in the circumstances to have revealed to him [or her]." *Id.* § 2-316(3)(b).

111. U.C.C. § 1-201(10).

112. *Id.*

113. 874 F.2d 653 (9th Cir. 1989), *amended* 890 F.2d 108 (1989).

114. 874 F.2d at 658.

115. 798 F.2d 299 (8th Cir. 1986).

116. *Id.* at 303.

117. 15 U.S.C. §§ 2301-2312 (1988).

118. *Id.* § 2308(a).

119. *Id.* § 2308(b).

120. Mislow, *Reducing the High Risk of High Tech: Legal Planning for the Marketing of Computer Systems*, 23 AM. BUS. L.J. 123,

137- 38 (1985). In *Micro-Managers, Inc. v. Gregory*, 147 Wis. 2d 500, 434 N.W.2d 97 (Wis. Ct. App. 1988), the trial court found that the software developer had "used its skill and expertise in good faith to design and develop the software," as it had promised its customer. 147 Wis. 2d at 517, 434 N.W.2d at 104. As a result, the appeals court held that the developer had substantially complied with and had not breached the contract, even though the customer complained that the software did not perform as expected. *Id.*, 434 N.W.2d at 104.

121. "[An] agreement ... may limit or alter the measure of damages recoverable under this Article, as by limiting the buyer's remedies to return of the goods and repayment of the price or to repair and replacement of non-conforming goods or parts" U.C.C. § 2-719(1) (a) (1989). *See also* *Earman Oil Co. v. Burroughs Corp.*, 625 F.2d 1291, 1298 (5th Cir. 1980).

122. 538 F. Supp. 776 (E.D. Wis. 1982).

123. *Id.* at 778.

124. *Id.* at 786.

125. *Id.* at 791.

126. U.C.C. § 2-719(2) (1989).

127. *Reece*, *supra* note 52, at 855.

128. "Where circumstances cause an exclusive or limited remedy to fail of its essential purpose, remedy may be had as provided in this Act." U.C.C. § 2-719(2).

129. 772 F.2d 543 (9th Cir. 1985).

130. *Id.* at 547.

131. *Id.* *See also* *Hawaiian Tel. Co. v. Microform Data Sys., Inc.*, 829 F.2d 919 (9th Cir. 1987) (contractual provision limiting consequential damages from failure of computer system not enforced when no system had been delivered at all).

132. The clauses provided in this Article are provided only as examples and should not be used without considering the needs of each situation.

133. This is authorized by U.C.C. § 2-719 (1989).

134. 686 F. Supp. 109 (D. Md. 1988).

135. *Id.* at 112 (emphasis in original).

136. U.C.C. § 2-718(1).

137. *Id.*

138. *Reece*, *supra* note 52, at 855.

139. A provision that requires careful attention is the limitation on consequential damages, especially as it relates to the remedies for breach of warranty. Any limitation of consequential damages clause should be separated from the clause limiting the remedies available upon breach of warranty; and it should be expressly applicable to a breach under any provision of the agreement, including the warranty provisions. If the limitation of consequential damages clause is construed to apply only to the limitation on warranty remedies, it can be invalidated by showing that the limitation on warranty remedies failed of its essential purpose. *See* U.C.C. § 2-719 (2) *and* *RRX Indus., Inc. v. Lab-con, Inc.*, 772 F.2d 543 (9th Cir. 1985); *compare* *Kearney & Trecker Corp. v. Master Engraving Co.*, 107 N.J. 584, 527 A.2d 429 (1987). In addition,, the United States Court of Appeals of the Ninth Circuit, has held that when the

limitation of consequential damages is part of the warranty disclaimer, the limitation does not apply to a breach of contract arising from a failure to develop and deliver the computer system at issue. *Hawaiian Tel. Co. v. Microform Data Sys., Inc.*, 829 F.2d 919 (1987).

140. "Consequential damages may be limited or excluded unless the limitation or exclusion is unconscionable." U.C.C. § 2-719(3). Note that there are different standards by which different types of plaintiffs may assert this barrier. "Limitation of consequential damages for injury to the person in the case of consumer goods is prima facie unconscionable but limitation of damages where the loss is commercial is not." *Id.*; compare *Collins v. Uniroyal, Inc.*, 126 N.J. Super. 401, 407-408, 315 A.2d 30, 34-35 (N.J. Super. Ct. App. Div. 1973) (per curiam), *aff'd*, 64 N.J. 260, 315 A.2d 16 (1974) (analyzing meaning of "prima facie unconscionable" in consumer context) with *Harper Tax Servs., Inc. v. Quick Tax Ltd.*, 686 F. Supp. 109, 112-13 (D. Md. 1988) (rejecting unconscionability argument in commercial context).

141. 127 Ariz. 278, 619 P.2d 1055 (Ariz. Ct. App. 1980).

142. *Id.* at 281, 619 P.2d at 1058.

143. *Investors Premium Corp. v. Burroughs Corp.*, 389 F. Supp. 39, 44 (D.S.C. 1974) (integration clause upheld; outside statements regarding the capabilities of the system held inadmissible); *Office Supply Co. v. Basic/Four Corp.*, 538 F. Supp. 776, 782 (E.D. Wis. 1982) (integration clause prevented admission of parol evidence to vary the terms of the agreement; however, integration clause does not affect implied warranties, which "will be held to exist unless they are specifically excluded.").

144. 874 F.2d 653 (9th Cir.), *modified, reh'g denied*, 890 F.2d 108 (1989).

145. *Id.* at 657, *modified, reh'g denied*, 890 F.2d at 112. For other cases in which vendors were held liable for inadvertent express warranties, see *Redmac, Inc. v. Computerland of Peoria*, 140 Ill. App. 3d 741, 489 N.E.2d 380 (1986); *Fundin v. Chicago Pneumatic Tool Co.*, 152 Cal. App. 3d 951 (1984); *Northern States Power Co. v. ITT Meyer Indus.*, 777 F.2d 405 (8th Cir. 1985); *discussed in Levy, supra* note 41, at 15-16.

146. Comment, *supra* note 9, at 848-50; *Reece, supra* note 52, at 854; *Sierra Diesel Injection Serv., Inc. v. Burroughs Corp.*, 651 F. Supp. 1371, 1377 (D. Nev.) (motion for reconsideration of magistrate's order), 656 F. Supp. 426 (evidentiary hearing) (1987), *aff'd*, 874 F.2d 653 (9th Cir.), *modified, reh'g denied*, 890 F.2d 108 (1989).

147. See *supra* note 132 and accompanying text.

148. See *supra* notes 61-62 and accompanying text.

149. *Kopf, supra* note 53, at 157.

150. U.C.C. § 2-725(1) (1989); *Mislow, supra* note 120, at 133. In general, a cause of action "accrues" when a breach of contract occurs, regardless of the buyer's lack of knowledge of the breach. U.C.C. § 2-725(2). There are separate provisions concerning suits for breach of warranty. *Id.*

151. *IBM v. Catamore Enters., Inc.*, 548 F.2d 1065, 1076 (1st Cir. 1976), *cert. denied*, 431 U.S. 960 (1977); *Cristo, The Applicability of Negligence and Malpractice to Data Processing Situations*, COMPUTER L. REP. 570, 576-77 (1983).

152. See *supra* notes 51-52 and accompanying text.

153. See also *Birnbaum, supra* note 37, at 153.

154. *Gregg & Folk, supra* note 60, at 22-23.

155. *Zaharoff, Expert Systems: Strategies for Minimizing Liability*, 6 COMPUTER LAW. 31, 31 (1989).

156. See *supra* note 2 and accompanying text.

157. Mislou, *supra* note 120, at 134-35.

158. "Any description of the goods which is made part of the basis of the bargain creates an express warranty that the goods shall conform to the description." U.C.C. § 2-313(1)(b) (1989); *see supra* notes 41-43 and accompanying text; *see also supra* note 145 and cases mentioned therein.

159. *See supra* notes 52-60 and accompanying text.

160. *See supra* notes 141-47 and accompanying text.

161. Zaharoff, *supra* note 155, at 33.

162. *See supra* notes 51-98 and accompanying text.

163. Sullivan, *supra* note 106, at 75.

164. *Cf.* Community for Creative Non-Violence v. Reid, ___ U.S. ___, 109 S. Ct. 2166 (1989) (establishing criteria for determining when a third party is to be considered an employee for the purposes of the U.S. Copyright Act).

165. Sullivan, *supra* note 106, at 75.

166. *Id.* at 77; Zaharoff, *supra* note 155, at 32.

167. Sullivan, *supra* note 106, at 78-79. An insurance company now offers computer virus coverage as part of its standard Electronic Data Protection policy. *Allstate Insurance Becomes First Company to Offer Computer Virus Insurance*, 1989 COMPUTER L. REP. 1019.

168. *See supra* notes 76-79 and accompanying text.

169. Zaharoff, *supra* note 155, at 31; Birnbaum, *supra* note 37, at 155-57.