MICALYN S. HARRIS

September 11, 2000

April M. Major, Esq.
Bureau of Consumer Protection
Division of Marketing Practices
Federal Trade Commission
Washington, D.C. 20580

Re: High-Tech Warranty Project -- Comment

Dear Ms. Major:

1. Thank you for your invitation to comment and for the opportunity to contribute to the public forum to examine warranty protection for software and other computer information products and services that are marketed to consumers, being held under the auspices of the Federal Trade Commission ("FTC" or the "Commission").

2. I am Vice President, Secretary and General Counsel of Winpro, Inc., a small, privately-held software consulting, design and development company, which provides both mass-market and custom software applications, and system design and implementation (programming) services. I was also an American Bar Association Alternate Advisor, Business Law Section, to the Uniform Computer Information Transaction Act ("UCITA") Drafting Committee, National Council of Commissioners on Uniform State laws, and actively contributed as an Alternate Advisor and an Official Observer at the Drafting Committee's open sessions. By way of responding to the specific questions from the Commission relating to UCITA, enclosed please find a copy of an article I wrote entitled, "UCITA: Helping David Fact Goliath", published in July, 2000.[1] This Comment has been reviewed for technical accuracy by the President of Winpro, Inc.[2] The views expressed are solely my own.

3. In its Notice, the Commission indicates that it is considering whether Magnuson-Moss-type warranties against defects in consumer products should be extended to computer software and other computer information and services acquired by consumers.

4. The Magnuson-Moss Warranty Act ("Act") was designed to deal with tangible products. The Act does not define "defect" but the defects consumers are most likely to encounter in tangible products are variations from the norm, e.g. manufacturing

---

[1] Micalyn S. Harris, "UCITA: Helping David Face Goliath", The John Marshall Journal of Computer and Information Law, Vol. XVIII No. 2, Winter, 1999.

[2] Dr. Louis J. Cutrona, Jr., Ph. D.

defects such as a faulty switch or arising from damages during shipping, such as a cracked faceplate.

5. When software is distributed, variations from the norm are rare, and when they occur, it is usually because of a defect in the tangible medium (diskette or CD-ROM) on which the software is distributed. All software providers of which we are aware warrant that the tangible medium on which the software is distributed is free from defects and offer to replace any defective media. In other words, consumers have, under current law, the type of warranty the Act was designed to provide with regard to the "product" aspects of the software or information.

6. The value of software and other computer information, however, does not reside in the value of the tangible medium and the "defects" of which consumers complain are not variations from the norm. Magnuson-Moss contemplates that "defective" products are not "merchantable". All software, however, is imperfect, that is, defective. Moreover, excluding defects in the tangible media on which they are distributed, whatever defects exist in a given software application or package, exist uniformly in all copies of that application or package. Because all software is defective and all copies are uniformly so, extending Magnuson-Moss-type warranties to software will give every consumer the right to sue every provider of software anytime that consumer wishes to do so. That result is likely to increase litigation, increase costs of software creation and commercialization, decrease competition in the industry, decrease the number of applications and information packages available to consumers, and increase the cost to the consumer of those which are available.

7. Attempting to hold an entire industry to a currently impossible standard by giving consumers the "protection" of the right to sue every software provider on the ground that its software is "defective" is likely to have serious adverse effects on the industry, on consumers, and on the national economy.

8. This Comment provides background information responsive to the Commission's questions and discusses some of reasons why an attempt to impose Magnuson-Moss-type warranties on software has a high risk of unintended adverse consequences and why, therefore, such an attempt appears ill-advised as a matter of public policy. The comment is divided into ten sections: (i) an overview; (ii) a brief history of the industry, (iii) background information on the legal status of software, (iv) a description of the legal and practical challenges which gave rise to licensing as the solution for the commercialization of software; (v) the current legal status of "shrinkwrap" licenses for mass-market software; (vi) a discussion of the effect of post-sale disclosure of licensing terms and the merits of the solution offered by the Uniform Computer Information Act ("UCITA"); (vii) a word about software and its complexity; (viii) a brief discussion of the Magnuson-Moss Act and why it is inappropriate to apply its underlying assumption that "merchantable" products are not "defective" to software; (ix) summary and conclusions; (x) a cross-reference sheet (enclosed after the last page of this Comment) indicating brief answers to the

Commission's 16 questions, and where, in this Comment, further elaboration is provided.

Overview

9. The mass-market computer software and computer information industry is in its infancy, and every aspect of the software and information it provides is changing more rapidly than in any industry we have previously experienced. Its growth has been astounding. The legal structure supporting that growth has been contracts, in the form of licenses. The licensing model offers the flexibility that is essential to meeting the needs of both providers and users. The risks of attempting to impose a different legal structure on the industry, e.g. by treating computer software and information as "products" which are "sold" rather than licensed, are potentially devastating. The potential rewards must be sufficient to make taking those risks appropriate. To date, we are not aware of any study indicating that substantial benefits are likely to be realized by attempting to substitute a new legal structure for that currently used throughout the industry.

10. Because the industry is relatively new - mass-market computer software and information has been a significant factor in the U. S. economy for less than 20 years, our experience with mass-market computer software and information is extremely limited. All of us would like to have perfect software, but so far, perfect mass-market software does not exist. Demanding, by imposition of mandatory warranties that cannot be disclaimed, that perfect, or even "better" software be produced, is likely to increase costs and reduce the range of applications available. The increased costs and burdens will fall most heavily on small companies. The result will be reduced competition, to the detriment of small companies, users of computer software and information, and society as a whole.

11. It is generally conceded by the public at large, by the academic community, and by the software industry itself, that there is virtually no mass-market software that is perfect. Under current law, each software provider determines what the software "should" do by advertising, and failure of the software to perform as advertised has the usual market and potential legal consequences: loss of reputation in the market, and consequent loss of sales revenues. Users who are damaged have the right to sue under federal and state anti-trust and false advertising laws. Imposing, by legislation or government fiat, minimum "standards" for computer software and information in the form of "warranties" which cannot be disclaimed, and which include imposition of consequential damages if the software is found to be "defective" will (because no software is perfect--i.e., all software is "defective") create a situation in which all software venders are in breach of such non-disclaimable warranties. Thus, all software vendors will be fair game for breach of warranty suits. Such a situation will: (i) discourage software production because of the increased risk of litigation, (ii) increase costs, (iii) decrease the ability of smaller companies to compete and (iv)

decrease the choice of available computer software applications and information. In short, it will decrease competition in the industry and thereby harm consumers.

Background - A Brief History of the Industry

12. In less than two decades, the mass-market, prepackaged computer software and information industry has burgeoned from tiny to in excess of $100 billion.[3] Of all those involved in the process of creating, marketing, and utilizing computer software, it is the small developers who are perhaps most eager to assure that whatever legislation is recommended or whatever regulation is imposed is fair and even-handed, because small developers are both providers and users, both licensors and licensees. They must rely on the rights granted in licenses for software they use as well as rely on the licenses they grant to protect their ability to commercialize their software applications. If additional burdens are to be imposed on developers, distributors and publishers, it is the small developers who are most likely to suffer from the weight of those burdens. There are thousands of these small developers. Many are individuals or small companies with fewer than a dozen people, who routinely deal with large companies. Together, these individuals and small companies provide the backbone of the computer information industry.[4]

Background: Software Is a Literary Work

13. By federal statute, software is a "literary work." As such, copyright inheres in software from the moment of its reduction to a tangible medium (which includes an electronic medium).[5] It has been argued that copies of software can (and should) be bought and sold just as copies of books (which are also literary works) are bought and sold. Books, of course, are not sold without restriction. Their "use" (copying or reprinting) is restricted to "fair use".[6] What constitutes "fair use" of books is not

---

[3] According to the United States Department of Commerce, the world packaged software market was $109.3 billion in 1996, of which $50.4 billion was in the U.S. The world packaged software market is expected to exceed $125 billion in 1997. U.S. Industrial Trade Outlook '98, United States Department of Commerce, 1998. Note that these figures reflect only *packaged* software. The Commission's Notice indicates a scope not limited to packaged software, and therefore, any regulations it promulgates would impact an even larger market.

[4] See Raymond T. Nimmer, *Images and Contract Law - What Law Applies to Transactions in Information,* 36 Hous. L. Rev. 1 (1999). With regard to the size and make up of the market for computer information, Mr. Nimmer notes:

> an image of routinely subservient purchasers (licensees or buyers) does not accurately reflect practice. The nature of the information marketplace accentuates the degree to which the inaccuracy exists. Most vendors of information who provide works to publishers are individual authors dealing with relatively large corporate purchasers. Although there are large companies in the modern computer software industry, the average size of a computer software provider is fewer than twelve employees. These small companies routinely deal with large corporate clients (purchasers). For example, Walt Disney Corp. is seldom the unsophisticated party, especially in the many contracts in which it acquire services from small software development companies.

Id. at 25 (citations omitted).

[5] See 17 U.S.C. Sec. 102.

[6] See 17 U.S.C. Sec. 107. "Fair use" is both a term of art, defined by federal law, and a colloquial phrase used in commercial discussions to describe what a grantor licensing the use of intellectual property thinks it

always obvious. Books have a long history and there is statutory and case law on the issue of what constitutes "fair use" of another's copyrighted work.[7] Nevertheless, what constitutes fair use continues to be the subject of litigation.

14. By comparison with printed books**,** the mass-market software industry is young**.**[8] What constitutes fair use of software is much less clear than what constitutes fair use of books.

15. Although software is, by federal law, a literary work, software is different from books in many ways. "Reading" software, which is done by machine, is different from reading a book, which is done by humans. Humans do not utilize software by "reading" it. Its function is to perform, and in order to perform, software must be copied into a computer's memory. Once copied, the computer "performs" by executing the instructions and manipulating the data that together comprise the software.

16. Copyright law forbids copying a literary work without permission, except for "fair use," which permits limited copying, as for example, copying limited portions of a book as part of a book review.[9] Because software must be copied in order to be useful (computers copy software in order to use it), permission to copy must be granted in order to commercialize literary works in the form of software. In some cases, e.g. commercial software code libraries, meaningful commercialization requires that the literary work be copied many times.

17. Early developers who wished to commercialize their software were, therefore, faced with the challenge of making it possible to permit people to use the software without requiring the developer to give up his copyrights. Developers (whether they undertook to publish and distribute their own software or arranged to have a publisher distribute it for a royalty or on some other basis) were aware that copying software is inexpensive and easy. Reprinting an entire book may well cost more than purchasing a second copy from the author or publisher. Not so software. Making a copy takes little time, and is substantially less trouble and less costly than licensing another copy. Mass-market license agreements advise the licensee of the circumstances under which copying is unauthorized, and therefore "unfair" because unauthorized copying violates the licensee's obligations to the licensor.[10]

---

is fair for the grantee to do with that intellectual property. The two uses of the term are not always congruent.

[7] See e.g., "Free Speech, Fair Use and the Joys of Sects: The Evolving Scope of Copyright Protection on the Internet", The DataLaw Report, November, 1996.

[8] Although personal computers, and some software, were available from Tandy and Apple, the market began to broaden and grow meaningfully only in 1982, with the introduction of the IBM PC**,** giving us less than twenty years of case law regarding these issues.

[9] 17 U.S.C. Sec. 107.

[10] "Fair use" is a term of art, and defined by federal law. See 17 U. S. C. §107. The term is also used descriptively, in the colloquial sense, to indicate intended use or use contemplated by the licensor as being within the scope of a grant of rights. For purposes of this discussion, "fair use" will be used in its colloquial sense unless it is described as "fair use as defined by federal law".

18. Developers were also aware (if only after talking with their lawyers) that if they did not make some effort to limit copying, their software might become "public domain." If this occurred, commercial potential would be destroyed.

Licensing - The Key to Commercializing Computer Software

19. Developers wanted to be able to provide their creations to a mass market. To do so, developers had to enable their customers to use the applications the developers created, while assuring the developers' ability to obtain income from their creative efforts. In other words, the developers' challenge was to find a way to facilitate intended use of software applications without sacrificing the possibility of commercial gain. The solution that the industry devised was licensing. The use of a contract between the developer (whether directly, or through a publisher who would undertake to distribute the software for the developer) and a licensee made it possible to tailor arrangements to particular applications, and to make changes quickly - an important element in the rapidly developing world of technology.

20. In essence, licenses tell customers what developers consider "fair use" of their "literary" creations. (Licensing has been used to protect other new technologies. When phonograph records first became available, they were licensed, and the licenses were written on labels on the back of single-sided records. When records had recordings on both sides and labels became smaller, the labels recited references to specific license numbers, e.g., "RCA Victor, licensed under Lic. No.__.")

21. Early software licenses restricted the licensee to installing the software on a single computer. When Borland amended its license to permit installation on several machines so long as the program was not used by more than one person at a time, it described the revised and extended restriction as permitting use of its software "just like a book."[11] The relaxed restriction quickly became the standard, but software providers continued to forbid installation on a network server which would make a program available to multiple users simultaneously. Later, program licenses permitted network server installation, but with the proviso that the user would purchase enough licenses to cover the maximum number of simultaneous users served by the server. The beauty of the licensing technique was (and remains) its flexibility - the ability to change rapidly to adapt to rapidly changing technology and a rapidly evolving market. Licenses were (and are) revised quickly to meet changing customer needs, demands and expectations.

22. The licensing technique also permitted commercialization of code libraries. A code library is typically purchased by a developer for incorporation into applications for third parties. By describing what could and could not be done with a code library, the licensor described what the licensor considered "fair use" of the application. Thus, the developer-licensor could, in effect, say to his developer-licensee, "You may put

[11] *See e.g.*, Borland's license for its Turbo-Pascal and C ++ software.

this into your applications for third parties, but they may not take the code library (either as a whole or its constituent units) out of the application you produce for them and put it into another application. If they want to do that, they have to come back to me and obtain a license, because while you have paid me to use this in your software creations, your customer has not. It's only fair that if your customer wants to use this other than in the application you provide, your customer has to pay me for that privilege."

23. In such a young and dynamic industry, the use of contracts to advise customers of what software authors see as fair use of their software literary creations has worked well. The industry has expanded, and the variety of software applications today is enormous and growing. Available software ranges from games for young children to systems that run manufacturing operations to applications used by sophisticated developers to create other applications.

The Current Legal Status of "Shrinkwrap" Licenses for Mass-Market Applications

24. Mass-market shrinkwrap licenses have repeatedly been held enforceable under basic principles of contract law.[12] If innovation and growth of an industry are an indication, the licensing technique has served the industry, and society, well. Competition in the software industry is keen. We have, as potential users, a choice of thousands of "off-the-shelf" software applications. License provisions have, to date, promoted, not prevented the creation and distribution of competitive applications.

25. Application of these basic principles of contract law means that license agreements are enforceable only to the extent other contracts in our society are enforceable. Thus, terms which are "unconscionable" or against public policy are not enforceable.

26. The theory underlying freedom of contract is that in a market society, supply and demand, the opportunity to obtain commercial benefits, and competition in the marketplace will assure that providers of goods and services will provide what people want and not waste resources on goods and services people do not want.

27. Where a software license is neither unconscionable nor against public policy, a statute, like UCITA, which calls for enforcing it in accordance with its terms, reiterates the common law of contracts. Such a law is a statement of confidence in the market system and the ability of the market to reject terms which are unacceptable.

---

[12] *See, e.g.*, *Adobe Systems Inc. v. One Stop Micro, Inc*., 84 F. Supp. 2d 1086(ND Cal. 2000); *Peerless Wall & Window Coverings, Inc. v. Synchronics, Inc.*, 85 F. Supp. 2d 519 (WD Pa. 2000). Other cases make it clear that where license restrictions are inconsistent with ownership of a copy, possession of a copy does not confer ownership of that copy, and restrictions on use of that copy are enforceable. *See e.g.*, *DSC Communications Corp. v. Pulse Communications, Inc.*, 170 F3d 1354 (Fed. Cir. 1999).

28. In the abstract, one might worry that individual "consumers," that is, licensees of mass-market software, have little "bargaining power" in connection with licenses. Experience indicates otherwise. When a license is insufficiently broad to permit a licensee's contemplated use, the would-be licensee either negotiates for what it needs or finds an alternative route to meet its needs. In general, developers and their distributors and publishers, both as users ("consumers", although possibly not as defined in Magnuson-Moss) and as providers, look to the basic principles of contract law, combined with the pressures of the marketplace, to facilitate commercial exploitation their software creations.

29. Developers, and their distributors and publishers, understand that unreasonable licenses have effectively "killed" commercialization of potentially lucrative applications. For example, several years ago, Borland International introduced a new version of its flagship software development application. Customers (in this case, developers) who acquired the new product quickly discovered that Borland had changed its license to provide that a developer would be permitted to distribute up to 10,000 copies of any software developed using the new application, but that for 10,000 copies or more, a separate license would have to be negotiated. Customers realized that if they used the application (as intended) to develop applications which turned out to be very successful, they would be "over a barrel" when they tried to negotiate to sell the 10,000th copy. They "wrote" to Borland on-line, many returned the application for a refund, and others refused to purchase it. Within six to 12 weeks, Borland posted a revised license, but a significant segment of the developer community found substitute applications or other ways to meet its needs, and Borland lost valuable market share which it never regained.

30. Thus, licensors are well aware that to the extent they over-reach, they risk losing significant commercial benefits. Evidence to date indicates that the market system works well, and that the current mass-market software market is large and highly competitive. To the extent competition is threatened, we have specific laws, i.e., the anti-trust laws, including laws prohibiting false, deceptive and misleading advertising,[13] to deal with the problem.

31. There is no evidence that the market is in need of regulation by way of having a uniform body of law dictating what may and may not be included in software licenses. Advocates for mandating inclusion of "consumer warranties" appear to be base their demands more on what they believe certain consumers would like included in the price of mass-market software than on what the market as a whole needs in order to be maintained as a competitive market.

Post-Sale Disclosure of Licensing Terms

32. In its Notice, the FTC expresses concern about the effect of post-sale disclosure of licensing terms, on the grounds that such an arrangement makes it "nearly impossible"

[13] Section 5 of the Clayton Act and Section 43(a) of the Lanham Act.

for buyers to compare the terms of competing publishers before committing to acquisition of a particular software application or software information package.

33. Under the standard rules of contract formation, the terms of a contract must be disclosed prior to entering into the contract, or the contract is either void *ab initio*, or voidable. Both the common law and Article 2 of the Uniform Commercial Code have, however, permitted enforcement of contracts originally entered into with open terms which are subsequently "filled in" by the parties**.** License agreements are generally too long to put on the outside of a software package. The result has been "shrinkwrap" and "clickwrap" licenses. These licenses provide a means for the customer to read the license before opening an envelope or shrinkwrap around the diskette containing the software, and usually provide that if the customer finds the license unsatisfactory, the unopened diskette may be returned for full credit.[14] The clickwrap provides a means for the customer to read the license before installing the software, and will abort installation if the customer does not agree to the license.[15] In these cases, if the customer has been billed prior to completion of installation, the customer is also generally permitted to return the software for full credit. In some cases, customers must contact the developer to obtain a return number; but the return of the diskette by mail is usually permitted, and usually a minor inconvenience (as compared with, for example, going in person to a retail store to return, say, a suit).

34. Even if all of the license terms were included on the outside of a software package, they would be available for review to relatively few customers. This is because distribution from retail stores accounts for only a few of the thousands of software applications made available to the mass market. Shelf space in retail stores is limited, and reserved for the most popular software packages, most of which are produced by the larger companies. In terms of the number of different software applications, the vast majority is distributed via catalogs, and ordered via mail, fax or telephone. For small developers, the most effective method of distribution is often a targeted mailing and fulfillment via mail, fax and telephone orders.

35. It is possible for retail stores to make license agreement provisions available to customers. When it had free-standing stores, Egghead Software maintained a file of software licenses for every package it stocked, and customers could review these licenses prior to making payment. One had only to ask. Egghead Software no longer has retail stores, which may indicate, among other things, that being able to review software licenses was not a significant competitive advantage or otherwise a matter of concern to customers.

---

[14] As a practical matter, a customer is permitted to return the unopened diskette for any reason, not just unhappiness with the terms of the license, and the provisions of the August, 1998 draft of 2B-208 reflect this reality.

[15] "Clickwrap" licenses may be included in packaged software, and are displayed at the beginning of the installation process. Software which is distributed electronically, that is, never packaged, also generally includes a "clickwrap" license to which a customer must agree before being permitted to download the software.

36. In any case, suppose there were a law that before shipping software, a licensor had to provide the customer-licensee with a copy of the license agreement and an opportunity to review it. Would the result benefit anyone? If so, who? Large software providers with packages available at retail would not suffer greatly. Retailers could be required to make licenses available to customers at the store. Catalog and mail or fax order sales however, would become considerably less efficient than they are now. Software distributed via catalog would become more expensive because prior to shipping the software, a copy of the license agreement would have to be sent out, and an acknowledgment of its acceptability linked up to the ordering customer. The channel of distribution most available to small developers, publicity through mailing lists and shipment based on orders placed by mail, fax or telephone, would become an administrative nightmare if a copy of the license agreement had to be sent first, and an acknowledgment of its acceptability received before software was shipped. The result would be to reduce competition for mass-market software, increase prices for the software which was available, and make it much more difficult for small developers to distribute software applications, thereby leaving the market to large providers. Such a result is far more anti-competitive than any possible anti-competitive impact arising from customers' inability to compare license provisions before receiving a shipment of returnable software.[16]

37. There is no evidence beyond unsupported allegations of certain "consumer advocates" that customers regard providing the specific terms of a license agreement prior to shipment as information worth paying for. Providing the information, particularly for catalog and direct mail sales, which are the primary methods of distribution available to smaller developers, has a cost which would have to be reflected in the price paid to them. Some academics and self-styled "consumer advocates" may believe that people choose software on the basis of what terms are or are not in a license agreement, but reviews rarely mention this factor as important. There is no evidence that the assumed benefit arising from preshipment review of license agreement terms would be commensurate with the additional costs and burdens, which are particularly onerous for the small developer.

38. If such a requirement were to be imposed, the likely result would be to favor Web-based distribution over catalog, mail-order and fax order distribution. In Web-based distribution, the provider can require the prospective customer to click through a license and agree to its terms before downloading a software application. This arrangement, however, has other problems. For a large program, and that includes most games, which are heavy with graphics, downloading can be an intolerably long

[16] If review of license terms prior to shipment is the objective, "clickwrap" arrangements for downloading from a distributor's Web site meet the criteria, as review prior to ordering is easily arranged. Grouping shrinkwrap and clickwrap licenses together is may fail to recognize this distinction. If, however, preshipment review of license terms were required, the requirement would support electronic distributors to the detriment of catalog distributors. As a matter of public policy, in the absence of discussion and evidence of a need for legislation to support one channel of distribution method to the detriment of another, it seems preferable for the marketplace, rather than government agencies or a legislature, to make that determination.

process. Setting up and maintaining a Web site involves development and administrative costs for a developer. Extra effort is required of a customer in ordering from a Web site, and some customers will be lost because of reluctance to place such orders. Perhaps more important, for small developers, mailings which require a prospective customer to order from a computer rather than by fax or telephone may be far less effective in terms of responses, thus decreasing the effectiveness of the most readily available and often least expensive route to sales revenue.

39. A more neutral and small-developer-friendly solution is offered by UCITA. That proposed law requires licensors distributing to consumers through channels which do not permit review of license agreements prior to shipment to accept return of shipped software - not just because the license agreement is unsatisfactory, but for any reason. This is a right which in other industries is granted at the discretion of the distributor, as a marketing tool, and not required by statute. However theoretically troubling the current "shrinkwrap" arrangement may be, permitting software to be shipped and returned unopened (for any reason, including but not limited to an unacceptable license agreement) is, particularly for the small developer, far more desirable than requiring preshipment disclosure, which has a cost that can only make software more expensive for developers to provide and therefore, more expensive and less appealing to users.

40. Destroying or significantly reducing the channels of trade for thousands of software applications in order to permit preshipment review seems, as a matter of public policy, a poor choice. Society does not impose such requirements on other items. If one brings home a shirt which turns out to be the wrong color, one may or may not have a right to return it, and in the absence of a defect, will have to bear the time and expense of a permitted return. It is not at all clear why software should be burdened with obligations not imposed on other items. As a matter of public policy, we do not, in general, insist that suppliers accept returns, or that they pay the cost of customer returns. Nevertheless, UCITA imposes that burden on software licensors if license terms are not available prior to payment and delivery of the licensed software. If there were any public policy concerns regarding inability to review license terms prior to shipment and payment, the UCITA solution, which requires a licensor who does not make license terms available prior to shipment and collecting payment, to place the licensee in as good a position as if he, she or it had had an opportunity to review such terms prior to ordering the software and found the terms unacceptable, is preferable to an inflexible requirement of preshipment disclosure.

41. Critical reviews of software, and there are many, indicate that the concern, perhaps, is not the absence of knowledge of license provisions, but the absence of particular provisions or services which certain consumers would like. For example, one frequent complaint is that many software providers charge for technical assistance. Providing technical assistance has a cost. A software provider can allocate that cost to those who use the service by charging for the service, or can allocate the cost to everyone who uses the software, regardless of whether they use the technical

assistance, by raising the cost of the software for everyone. Requiring that a certain number of hours of technical assistance be provided with all products effectively requires software providers to allocate the cost of such assistance across all of the users of their products, rather than to allocate the cost exclusively to those who use the service. No rationale has been offered for the proposition that it is appropriate for legislation, rather than the market, to determine whether technical assistance is required, and who should pay for it. Insisting that a rule or regulation (particularly in the absence of a law) require that technical assistance be provided is likely to impose the greatest burden on the small developer who is least able to bear that burden. Such a requirement would favor large providers who have many products, and would place what may be an insurmountable burden on small developers who have only one or a few products, which may or may not be able to support the minimal amount of required "free" technical assistance. Such a proposal, if adopted, would be enormously burdensome to the many small businesses which have developed and are attempting to commercialize software.

A Word About Software: Amazing, Wonderful, Complex, and Defective

42. Forty years ago, only the military and a few large institutions had computers. Everybody who worked with them was an expert. Twenty-five years ago, microcomputers began to be available. Eighteen years ago, IBM made its first personal computer ("PC") available, and mass-market software was on its way to becoming a significant segment of our economy.

43. Software is complicated. It's a wonder that any of it works at all. If we required software to work with the reliability of a 747 jet airplane, we should expect the cost of software to increase correspondingly. The analogy to a jet airplane is not inappropriate. A medium-sized computer software program has about 100,000 lines of code and is comparable in complexity to a 747, which has about 100,000 parts.

44. The 747 jet airplane took over three years just to design. It probably required a dozen or more people for the design, and hundreds of people to build. If 100 people worked on building it, each would have had, on average, responsibility for 1000 parts. Testing was extensive, and did not eliminate all of the defects.

45. Our company recently completed an educational simulation to supplement the teaching of algebra to middle school students. The simulation has about 100,000 lines of code. Our 100,000 plus line simulation was produced over a period of approximately 12 months, including two months of testing. The staffing for it consisted of seven people: a producer, an instructional designer, two writers, a graphic artist, and two programmers. The application will be available to thousands of educational institutions. And it will cost each of them a lot less than a 747. It is also worth noting that if it malfunctions, no one is likely to die or be seriously injured. The investment which is appropriate for testing a jet airplane is not, in economic risk-reward terms, appropriate for testing most software. When that kind of testing is

appropriate, as for example, to test software which runs a heart-lung machine, the cost of the software reflects the cost of that additional testing.

46. All software is intrinsically "buggy", or, more formally, defective. Software is complicated. It is doable only because it involves working with intangibles. It is not necessary to physically handle or machine each part. Software suppliers and everyone who uses software know it is defective. Adobe makes available lists of known "bugs" that is, defects in its software, and is highly regarded for doing so. Information regarding these defects ranges from descriptions of "fixes" or "workarounds" to messages of the import, "under certain conditions, such-and-such will occur. We don't know why it occurs it but it does. We don't know of a fix." Despite such defects, the software is regarded as not merely useful, but enormously valuable.

47. Windows 2000 has over 50 million lines of code. Are there defects in Windows 2000? Unquestionably. Some defects are known, for some there are fixes, Microsoft is no doubt in the process of fixing some of the defects, some of the defects Microsoft will decide are not worth fixing, and some of the defects have not yet come to light.

48. Most mass-market software providers try to the responsible. Intuit, which provides TurboTax, maintains an automated facility enabling users to download, via the Internet, the latest fixes, changes and updates. The arrangement permits Intuit, when necessary, to replace arbitrarily large portions of the code on the CD-ROM on which the program is distributed. There are still "bugs". Intuit knows there will still be "bugs". Intuit also advertises "100 percent accurate calculations" and backs its promise with the following guarantee: "If you pay an IRS or State penalty because of a calculation error in TurboTax, we'll pay you the penalty plus interest."[17]

49. Where there are problems which providers can reasonably foresee, and reasonable steps they can take to eliminate them, providers are willing to take those steps, if only to protect their revenues. Most software licenses state that the software operates "substantially in conformity with the documentation." That standard, as well as standards set by a firm's advertising, are standards with which providers can deal. Standards established by "consumer expectations" are vague and may not be either ascertainable by a provider or possible for a provider to meet. When software does not function as represented in the governing license agreement or in advertisements, a licensee has recourse by way of an action for a breach of contract or for false advertising and/or unfair trade practices under Section 5 of the Clayton Act, Section 43(a) of the Lanham Act, and corresponding State statutes. Consumers, and others, who have legitimate grievances can bring wrongdoers to court under existing laws.

50. All software is defective. Giving consumers the right to sue software suppliers if the software is "defective" will give consumers the right to sue software providers whenever they feel like it. There is no evidence that holding an entire industry to an impossible standard will benefit consumers. To the contrary, the most likely outcome

[17] *See e.g.*, back panel of box for TurboTax 1998.

will be to drive small developers out of the business of developing for the mass market. The cost of providing software will increase, the range of software available to consumers will decrease, the cost of the software that is available will increase, and the vitality of the industry will be sapped.

The Magnuson-Moss Warranty Act

51. As the Commission notes in its Invitation to Comment, the Magnuson-Moss Warranty Act ("Act") was passed in response to "problems encountered when the products they (consumers) purchased which were defective." The Act applies to the purchase and sale of tangible personal property by consumers. It does not define "defective". It assumes that the word has a well-understood meaning and that there are established standards and sufficient experience to determine, in a particular case, what makes a product "defective" and what makes a product generally "acceptable" and therefore "merchantable". The Act assumes that "merchantable" products are not "defective".

52. The Act does not define "defective." The Oxford English Dictionary ("OED") defines "defect" in several ways, which may be seen as moving along a continuum of acceptability from (i) "absence of something essential to completeness" to (ii) "a shortcoming or failing; a fault, blemish, flaw, imperfection". A person, or a software application, need not be without fault, flaw or blemish in order to be well-regarded and valued. As the OED notes, "The very good general reputation he had, notwithstanding his defects, acquired" citing 1647, Clarendon, *Hist.Reb.*1 (1843) 25/I.

53. The Act assumes that consumers need protection from physical defects in particular physical "copies" of "merchantable" physical products that vary from a norm, and that the norm does not have these defects. Software is a literary work. If I buy a novel and my copy has blank pages where Chapter 10 is supposed to be, I can exchange it for a complete copy, but if the copy is complete and I don't like the ending, I am not entitled to return the book. Software performs. If I attend a stage play, and the cast decides not to perform the last act, I may be entitled to a refund, but if the entire performance is poor (defective in my view), or if I don't like the play (so poorly written as to be, in my view, defective), I am not entitled to a refund, or to see a better performance. When a consumer complains about "defects" in software, it usually means that the software is either doing something the consumer doesn't want, or not doing something the consumer does want. It is, in other words, a complaint about performance.

54. In the world of physical products, situations in which an entire industry produces products that are defective are relatively rare (asbestos comes to mind, and possibly, tobacco products). When defects are dangerous, such as tires that fall apart at highway speeds, recalls can be mandated if a supplier is unwilling to take recall action to protect its reputation and minimize potential risks and damages. All software, not just consumer software, is defective, but it is not physically dangerous. The Act's

basic assumptions that most products of a given type are not defective and that "merchantable" products are not "defective" do not reflect the reality of software.

55. Magnuson-Moss warranties were never designed to encompass intangible property or intellectual property which is commonly licensed. Treating computer software and information as "tangible personal property" flies in the face of reality (because its value is in the performance, not the tangible medium). An attempt to extend the Act to include intangibles is likely to be criticized and challenged successfully. Thus, if Magnuson-Moss-type warranties are to be imposed, legislation to that effect would be required. No detailed rationale explaining the need for such legislation has been articulated.

56. Imposing Magnuson-Moss-type warranties on computer software and information is likely to effectively destroy the ability of small developers to commercialize their creations for mass-market consumers. Because all software is defective, a substantial body of litigation would be required to establish standards distinguishing between acceptable and unacceptable defects. Big companies could afford such litigation burdens and distractions. Small companies could not. Most small developers would simply turn their talents to providing programming services to larger companies that could afford to take the added risks and protect the resulting creations with litigation if necessary or deemed desirable.

57. To summarize: all software is defective. Nevertheless, it is for the most part, so far superior to the alternative of being without it, that we tolerate its defects. The Act is conceptually inapplicable to software because it assumes that if a product is "merchantable" it is not defective. That simply is not the case with software. Software is both "merchantable" and "defective."

58. Requiring, by law or regulation, that software not be "defective" will not improve the quality of software. On the contrary, it will make software more expensive to create, and will eliminate a great deal of software which is now available. Why? Because providing perfect software is so far impossible. Providing better software (whatever that means in a particular case) has a cost. Whether that cost is more time required for creation, or more time devoted to testing, or more time spent in litigation defending the acceptability of an application's imperfections, increased costs will result in more time and cost being required to produce a particular application or information package. The burdens imposed by that additional time and cost will fall most heavily on small developers, who will also face increased financial risks of providing software, because all of it will be imperfect and therefore subject to claims for "damages" from "defects". Rather than undertake such burdens, and face smaller potential markets because the law of supply and demand says that higher prices will result in decreased demand and therefore a smaller potential market, small developers are likely to turn their talents to creating non-consumer applications, or to go to work for large organizations which can afford the additional costs and burdens more easily. This will decrease the number of competitive suppliers, decrease the number of

programs available, and increase the cost of those programs. Decreasing competition in the industry, decreasing choice, and increasing costs is not beneficial either to consumers or to society as a whole.

Summary and Conclusion

59. The industry currently gives Magnuson-Moss type warranties with regard to the "product" portion of software, that is, the tangible medium on which it is distributed. There is no evidence that the software industry licenses are vague or extremely technical or difficult to understand and interpret.

60. To the extent that abuses exist in the software industry, remedies exist under current law. For example, to the extent that software publishers fail to provide what they promise, the remedy lies in a breach of contract claim, or possibly, an action for fraud in the inducement, in which case the contract is unenforceable. If advertising is false, deceptive or misleading, remedies are available under Section 43(a) of the Lanham Act and corresponding state statutes prohibiting unfair and deceptive trade practices.

61. There is no evidence, and no rationale has been articulated, for the proposition that it is necessary or appropriate for federal law to dictate the terms of software license agreements. Under current law, and the proposed UCITA, where there is a demand for more advantageous terms, offering them could be used as a marketing tool. In a market system, if additional grants of rights provide an economic return, such grants will be used as marketing tools to provide a competitive advantage. Where there is no demand for such grants, or when the market does not value them sufficiently to cover the cost of providing them, they will not, in a market-driven economy, be provided. Particularly in an industry in which technology and needs change rapidly, attempting to determine, by statute, what will constitute a desirable allocation of resources by dictating what rights must be granted is unlikely to result in maximizing the economic return on the resources utilized in that industry.

62. Where the market system is working, or other avenues for appropriate government intervention exist when it is not working, permitting proposed legislation to become a basis for negotiating more favorable terms for certain interests than they can now obtain in the market place is, at least in a market system, ill-advised. Such negotiation through legislation is particularly undesirable when these interests make demands without regard to their second-level effects, which frequently take the form of unintended (and therefore unconsidered) consequences of government intervention.

63. If there are reasons why consumers need the "protection" of giving them the right to sue every provider and distributor of software, those reasons deserve detailed articulation so they can be examined and the costs and benefits of imposing such warranties evaluated. Mere assertion of the fact that software is defective and often leaves its users frustrated is not sufficient.

64. Extending Magnuson-Moss-type warranties to consumer software will not result in better software. Competition is creating better and better software, at lower and lower prices, or at comparable prices for more and more powerful applications. Imposing Magnuson-Moss-type warranties on computer software and information will have unintended adverse consequences. It will reduce the number of competitive suppliers, reduce the number of software applications and the amount of software information available to consumers, and increase the cost of those applications which are available, to the detriment of small developers, consumers, competition in the industry, and society as a whole.

65. The Commission's initial public forum is an ideal setting in which to explore the exact nature of consumer complaints regarding software, articulate those complaints, examine whether the complaints simply reflect frustration with software's imperfections and if so, whether imposition of warranties is likely to remedy the complaints, and if so, at what cost. In reviewing these issues, it is appropriate to include representatives of small developers, and to weigh their interests and concerns, as it is small developers who make up the backbone of the industry. If I can be of additional assistance in connection with this process, please feel free to contact me.

Very truly yours,

Micalyn S. Harris

Enclosures (2)

Cross-reference sheet
http://www.adr-ny.com/articles/FTCComment-CrossReferenceSheet.pdf

"UCITA: Helping David Face Goliath"
http://www.adr-ny.com/articles/UCITA-HelpingDavidFaceGoliath.pdf