

**BY THE WAY,
IS THAT A GOOD OR A SERVICE?
SOFTWARE IMPLEMENTATION
PROJECTS LITIGATION:
CONTRACTS WITH A TECHNO TWIST**

SC Bar Continuing Legal Education

Marcus A. Manos
Partner
Nexsen Pruet Jacobs & Pollard, LLP
1441 Main Street, Suite 1500
Columbia, South Carolina 29201
(803) 253-8275
mmanos@NPJP.com

Presented at a Seminar sponsored by the S.C. Bar on September 22, 2000
University of South Carolina, School of Law

Software: *n* 1 : The entire set of programs, procedures, and related documentation associated with a system and esp. a computer system; *specif*: computer programs...

Litigate: *v* to carry on a legal contest by judicial process.

Webster's New Collegiate Dictionary, 1974.

INTRODUCTION

Software continues to grow as a part of everyday American life and the economy. Tech companies lead the surging bull market, and none of their products or services work without advanced software. No office worker in America today does their job without software. Many pundits predict the future of consumer purchases lies on the Internet. No one accesses the Internet without software. As the potential value of software continues to geometrically increase, disputes over who owns the rights to software will become even more common place. Businesses and individuals rely more on software for productive tasks, consumption and recreation each day. The increased use and expectations leads to more disputes over whether or not software performs as represented or agreed.

Courts on both the state and federal level now face software issues on a regular basis. As with most developing areas, the courts seek to apply traditional rules of law and then carve out exceptions if necessary. Legislators on both the state and federal level recognize the value of software and have made attempts to harmonize and structure the law as it applies to software products.

This presentation will give you an understanding of software implementation project litigation, how it tends to develop in the courts, and steps you can take to maximize your litigation position. Software litigation usually falls into three areas. This presentation focuses only on the first, and exclusively on a subcategory of commercial software implementation projects. Be aware of the other two, however, as the issues may come up in the implementation project case.

First, the purchaser, lessee or licensee of the software sues because the software allegedly does not perform properly. This litigation will be decided using contract law principals, possibly controlled by the Uniform Commercial Code, including warranty law. In this day of shrink wrap licenses, click wrap licenses, and sales of software on the Internet, consumer protection law and principals of personal jurisdiction as related to electronic commerce can also be important concerns for a software developer. Software performance litigation comes in two distinct types--the large purchaser, lessee, licensee who gets a custom or modified product as opposed to the "mass" business or consumer application.

The second type of software litigation involves a purchaser/lessee/licensee violating its agreement or maintenance agreement with the developer. A violation can involve something as simple as non-payment. More complex legal issues arise when a customer makes unauthorized modifications, allows unauthorized access, makes unauthorized copies, or sells the software package itself or maintenance/modification packages for the software. This litigation will also rely on contract principals, including any extraordinary relief provided for in the contract. Software misuse litigation also often involves copyright, patent or trade secret law.

The third type of software litigation involves a competitor or other person incorporating a portion of your code in his product, making/selling unauthorized copies, making/selling

unauthorized modifications or otherwise unlawfully displaying, using, selling or manufacturing your software. These software piracy cases do not involve contracts at all. The deciding legal principals will come from federal copyright and patent law and, to a lesser extent, from state or federal trade secret law. The Business Software Alliance (BSA) and Software Piracy Association (SPA) both pursue this type of litigation on behalf of their members. The BSA and SPA may also pursue a customer who makes unauthorized copies but likely will do so only on copyright or patent grounds rather than contract grounds.

The ubiquitous spread of software brings about software litigation. If you look, however, in the Westlaw or Lexis databases for legal research at the Fourth Circuit, Fourth Circuit District Courts and South Carolina Courts, you may be surprised at what results from a search for “computer” in the same sentence with “software.”

Not unexpectedly, copyright infringement, patent infringement, trade secret misappropriation, and unfair competition cases are common. Disloyal employee cases also appear frequently. Cases about the actual contracts to purchase software do not appear frequently. Indeed, at the time this research was done, not one such case came up in the South Carolina appellate courts under the search.

Software contract litigation often involves large sums of money or vital business functions. If the developer can enjoin the use of a vital program, a business could be destroyed. If a company spends millions on a development project and receives a program that cannot achieve its business goals, count on the shareholders taking notice.

Software implementation litigation almost always springs from the definition of what the software must do. One expects businesses to provide detailed charts of the flow of data, business

requirements, and technical parameters that the software should meet. When this happens, the litigation team has its road map before it. Surprisingly, the parties to a development project often define the software in very broad terms, (i.e. “loan application data processing and storage system”), or by reference to an outside document or source which is in flux, (i.e. “with functions similar to software x for small business”). Determining what the software should do is the first step. Then you must find the evidence which establishes the functionality. Finally, you must determine if that crucial evidence is admissible in the face of the contract.

The outline below and the discussion will point out common issues and problems which result in software implementation litigation and, hopefully, provide you with some helpful hints.

OUTLINE

I. Themes, Investigation and Pleading

1. Thematic development: These cases do not differ from any other litigation—selecting, developing, and presenting a few clear, understandable themes is essential to a winning strategy. Two focus points usually develop most or all of the themes.
 1. Does the software do what it is supposed to do?
 2. Did the developer make misrepresentations to sell the purchaser on the project?
2. Early investigation must focus on these two matters. Software implementation projects are fluid—the target tends to move. Find out who changed the requirements and why, did this result in delays? What was the starting point of the project—these are all key elements.

3. The contract(s) are key. The contract(s) will likely contain draconian limitations on remedies and claims. (See sample provisions at **Attachment 1**). The contract may very well contain forum selection, venue or choice of law provisions. Know these provisions and make an educated determination as to whether or not you want to avoid them. **An essential area of inquiry is the defined functionality and performance criteria for the project.** Search for all evidence that has an impact on what the software was supposed to do, how fast, with what interfaces, with what reporting capabilities, and **when** it was to be implemented. The most basic level of conflict in these cases is did the software do what it was supposed to do with the speed and methodology agreed on by the parties and was it implemented timely. Remember these projects usually result from some form of Request for Proposal type presentation by several vendors—even if fairly informal. There may be consultants on both sides to be interviewed.
4. A major battle field will be the scope of the litigation—restricted to the contract, or will misrepresentation, unfair and deceptive trade practices and tort issues come into play.
5. Software implementation projects usually do not break down quickly—despite the litigation positions you may end up taking, both the developer and the purchaser usually have major amounts of time and money invested and often the purchaser will have a pressing business need. If you are lucky and your client involves you early in the break down process there will be key decisions points that could determine who is the plaintiff and where the litigation will occur.

1. The purchaser will probably have an obligation to notify the developer of defects in the software and provide an opportunity to cure. If the purchaser believes the project is not remediable then the purchaser may want to file suit when it sends the cure letter.
 2. Each side probably will have an obligation to terminate the contract. Sending a termination letter without filing a lawsuit is often an invitation for the other party to file first.
6. Characterization of the parties.
1. The developer will want to characterize the project as a joint venture, show the customers computer expertise and experience, the joint development effort.
 2. The purchaser will want to show that it relied on the developers superior expertise, had a limited opportunity for due diligence and expected the product to be delivered ready to “turn the key.”
7. Pleadings.

Sample complaints are at **Attachment 2**. I have also attached with permission an article from the ABA publication *The Brief* which is very helpful as **Attachment 3**. The purchaser will normally seek to plead any theory that can set the contract aside such as fraud in the inducement, mistake, unconscionability, or other theory that the facts will support that will allow it to escape the limitations typically found in the contract. If the developer is bringing the action seeking payment/royalties it should carefully evaluate any extra contractual theories other than unjust enrichment or *quantum meruit*. The developer can almost always expect a non-performance

counter claim and if the developer has gone outside the contract, it will be much easier for the purchaser to do so. The purchase should plead any theory supported by the facts that allows for punitive, treble, or attorneys' fee damages. It increases the risk of the litigation for the developer and, therefore, increases the likelihood of settlement.

Sample answers and counterclaims are at **Attachment 4**. Many different affirmative defenses may apply. Look for facts supporting novation, accord and satisfaction, modification, waiver, estoppel or other defenses. Remember, these parties will in all likelihood have a long history of negotiation and performance that may include key changes in the contract or legal positions without the highest level of the company even knowing it. Sample replies to counter claims are at **Attachment 5**.

8. Forum selection. Federal versus state court. Be sure to use Complex Case Designation in Circuit Court. Form at **Attachment 6**.

NOTE: You do not have to be a programmer or computer literate to try these cases. You do need a full understanding of how software is developed, the terminology used, and a good expert to help you. If the case will support the cost, I always suggest a consulting expert hidden from your opponent as well as your testifying expert or experts. Terms like development tool, GUI, database, browser, C++, SQL, operating system, source code, object code, hardware, requirements tracing, project plan, unit testing, system testing, acceptance testing, etc. must have meaning to you.

II. Discovery.

1. Look for the consultants, independent contractors and sub contractors. Software implementation projects are rarely negotiated, contracted, programmed, tested, and implemented by the parties themselves. Usually there are consultants and

independent contractors involved. The contract lawyers great care to make sure that the sub or independent contractor or consultant cannot bind the principal etc. etc. may work strongly in your favor in terms of whether there is any privilege attached. Also, these independent contractors and consultants are much more likely to record something negative about the principal than the principal's long term employees.

2. Electronic information. You will be dealing with a software developer and the information systems department of its customer. Everything will be in electronic form as well as hard copy. Most if not all communication will have been by electronic mail. You must go after this information aggressively. Look for back up both on individual machines, servers, networks and stored on CD-ROM, tape or diskette. If the opponent says there are none—pursue. This is rarely the case. The information exist somewhere, you may even need to hire a reconstruction specialist to help you pull the information out of systems limbo.
3. Protective orders. One or both parties will certainly claim that the information involved is a trade secret or confidential, sensitive business information. Work out a protective order early to avoid a fight. If the opponent does not cooperate, put forward a reasonable protective order on your own so you can show the court that you have been reasonable when the motions fight comes.
4. Use your consulting expert to help prepare for depositions.
5. This is a case about high tech issues, use high tech. Get video tape depositions, use split screen and projector technology so exhibits can be shown with the testimony on the same screen.

6. Sample requests to produce are at **Attachment 7**. Sample interrogatories are at **Attachment 8**.

III. Selection and use of experts.

As with any technical, complex case the selection of testifying and consulting experts is critical. The Computer Science Departments at Clemson, USC, UNC, Duke and NC State have all helped me extensively. I have also used Sue Minter of On-Line Systems, Inc. to help in selecting a testifying expert, to assist as a consulting expert, and as a testifying expert. Your expert must have good credentials. I prefer someone who has testified a few times but does not do litigation support as a primary business focus. In other words, I believe an expert with a primary business of programming or consulting is more persuasive than one who spends the majority of his/her time in court.

Be sure to use both your clients and your own in house expertise. Most law firms and certainly your client will have an advanced Information Systems Department. These folks can help you immensely in organizing and understanding the litigation.

IV. Motions' Practice and key cases.

1. Motions will play a large role in the shape of trial. The developer will move to dismiss or for summary judgment on the non-contract claims. The purchaser should move for summary judgment or to dismiss many of the affirmative defenses and counterclaims. The outcome of these motions can affect the value of the case. Consider this in deciding when to mediate. See Foundation Software Laboratories, inc. v. Digital Equipment Corp., 807 F. Supp. 1195 (D. Md. 1992); Investors Premium Corp. v. Burroughs Corp., 389 F. Supp. 39 (D.S.C. 1974).

2. The developer can employ a very effective weapon if it sues or counterclaims for breach of the agreement by seeking a preliminary or temporary injunction to prevent the purchaser from using the system. In cases where the purchaser is using the system to “limp along” the granting of such an injunction can bring the purchaser to the table ready to settle immediately. A sample motion is at **Attachment 9**.
3. Contract law principals will control unless fraud in the inducement is established. See Scheduled Airline Traffic Offices, Inc. v. Objective, Inc., 180 F.3d 583 (4th Cir. 1999) (Virginia law); Management Systems Associates, Inc. v. McDonnell Douglas Corp., 762 F.2d 1161 (4th Cir. 1985) (North Carolina law).
4. One vital question that comes up consistently is the application of the Uniform Commercial Code (“UCC”). Most early cases found that custom software programming constituted a service not covered by the UCC. More recently, several courts have held that custom programming for business solutions, especially when combined with the sale of hardware, is more in line with a sale of goods and governed by the UCC. See Hartford Mutual Ins. Co. v. Seibels, Bruce & Co., 579 F. Supp. 135 (D. Md. 1984); also see K. Carson & G. Horowitz, “Software and Computer Law: Old Questions to be Answered in the New Millenium.” 10 Boston Bar Journal 43 (Oct./Nov. 1999).
5. Keep your research updated. A proposed Article 2B of the UCC for Software Contracts and Licenses of Information is working its way through the Uniform Acts process, **Attachment 10**, and has evolved into a Uniform Computer Information Transaction Act is lurking in the wings. See, Practicing Law Institute, “The UCITA

Revolution, The New E-Commerce Model for Software and Database Licensing.”
PLI Order No. GO-0009 (April/May 2000) (Available on Westlaw); J. Chanin, “The
Uniform Computer and Information & Transactions Act: A Practitioner’s View,” 18
J. Marshall J. Computers & Info. L. 279 (Winter 1999).

V. Trial.

1. You must have simple, understandable examples of how the software does/does not work. The jury must first understand the purpose of the software and then be shown the material deficiencies or, on defense, that it works. If the software is to issue checks, show it does not. Do not lead with complex technical failures, save them for the middle from your expert. IF YOU ARE REPRESENTING THE PURCHASER, your client must clearly testify about how the failure of these simple examples keeps the system from achieving its business purpose.
2. Use diagrams, charts, videos and animations. Spread them out. You will probably have to present deposition testimony, use video when possible or a lively presenter if you have to read.
3. Drive home the themes.
4. Your expert must speak sixth grade English, not “web speak” or “hackerese”.
5. Use summaries and demonstratives to guide the jury through the huge blizzard of paper.

CONCLUSION

Software implementation project litigation involves complex factual issues which require a detailed technical understanding of software programming, software interaction, testing, and system

implementation issues. The projects generate a large volume of electronic information and hard copy documents. Normally the projects involve a fairly large number of witnesses as well. As a result, clients look for lawyers with experience in this area and are willing to pay premium rates for assistance. As stated earlier, the lawyer need not have a programming or computer science background, but must be familiar with recent technical literature and terminology. It is extremely helpful to have personnel in your own law firm information systems department who have experience with programming, system implementation, and litigation support, as they will know all the issues from both the technical client side and from the legal side.

The legal issues presented by these cases generally fall within traditional contract and business tort doctrines. While the legal issues are not novel, they are complex in and of themselves as it relates to when can an action in tort and contract be brought at the same time versus how can the contract be avoided. Choice of law, applicability of the UCC, and the applicability of contract provisions restricting rights and remedies are all initial legal barriers which must be dealt with in every one of these cases.

All that being said, this litigation is entertaining, profitable, and interesting. With the continuing expansion of computer technology in business and personal lives, we can expect this type of litigation to remain a regular issue on the state and federal court dockets well into the future.

TYPICAL CONTRACT PROVISIONS

Here is a little taste of what you will encounter in the world of software contracts.

A. LICENSOR WARRANTIES AND REPRESENTATIONS.

Licensor represents and warrants that:

6.1 As of the Agreement Date Licensor has disclosed to Licensee in writing any and all known programming errors and operational deficiencies with the Disaggregated Software.

6.2 Licensor has title and right to convey the Disaggregated Software and Licensee's Rights to Licensee.

6.3 The Disaggregated Software does not infringe any patent, trademark, copyright, trade secret or other proprietary right of any third party.

6.4 LICENSOR MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, WITH REGARD TO THE DISAGGREGATED SOFTWARE, OBJECT CODE, SOURCE CODE AND LICENSEE'S RIGHTS. LICENSOR EXPLICITLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR A PARTICULAR PURPOSE. FURTHER, LICENSOR EXPRESSLY DOES NOT WARRANT THAT THE DISAGGREGATED SOFTWARE, OBJECT CODE, AND SOURCE CODE WILL BE ERROR FREE, WILL OPERATE WITHOUT INTERRUPTION OR WILL BE COMPATIBLE WITH ANY HARDWARE OR SOFTWARE OF THE LICENSEE.

6.5 Licensor provides the Disaggregated Software AS IS with no warranties, (except as expressly provided herein), or guaranties of performance.

B. LICENSEE WARRANTIES AND REPRESENTATIONS.

Licensee warrants and represents that:

7.1 Licensee is in the business of programming, evaluating and selling software for use in dental offices and that Licensee has expertise in the programming and evaluation of software such as the Disaggregated Software.

7.2 Licensee has had the opportunity to fully inspect and test the Disaggregated Software and to evaluate the Disaggregated Software prior to the Payment Date, if any.

7.3 LICENSEE MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, WITH REGARD TO THE SUBJECT MATTER OF THIS AGREEMENT.

C. LIMITED LIABILITY.

EXCEPT FOR THE PARTIES' INDEMNIFICATION OBLIGATIONS UNDER ARTICLE 9, NEITHER PARTY SHALL BE LIABLE FOR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE. THE MAXIMUM LIABILITY UNDER ANY LEGAL THEORY, CLAIM, OR CAUSE OF ACTION (i) OF LICENSOR TO LICENSEE SHALL BE THE LICENSE FEE PAID TO LICENSOR; AND (ii) OF LICENSEE TO LICENSOR SHALL BE ANY LICENSE FEE DUE AND UNPAID TO LICENSOR.

D. INDEMNITY.

9.1 Licensor shall indemnify, defend and hold Licensee harmless from any claim brought by any third party based upon the infringement of any patent, trademark, copyright, trade secret or other proprietary right of any third party in the Disaggregated Software as it is delivered to Licensee hereunder.

9.2 Licensee shall indemnify, defend and hold Licensor harmless from any claim brought by any third party based upon the infringement of any patent, trademark, copyright, trade secret or other proprietary right of any third party in the Disaggregated Software as modified by Licensee after it is delivered to Licensee hereunder.

IV. WARRANTIES.

4.1 Seller provides the Disaggregated Software **WHERE IS AND AS IS** with no warranties or guaranties of performance.

4.2 Purchaser represents and warrants that it has had the opportunity to fully inspect and test the Disaggregated Software and to evaluate the Disaggregated Software prior to the execution of this Agreement. Purchaser warrants and represents that it is in the business of programming, evaluating and selling software for use in dental offices and that Purchaser has expertise in the programming and evaluation of software such as the Disaggregated Software. Purchaser represents and warrants that it has determined the Disaggregated Software is satisfactory for the uses intended by Purchaser.

4.3 Purchaser and Seller warrant and represent that they understand this sale to be final.

4.4 Seller represents and warrants that it has title and right to convey the Disaggregated Software and Purchaser's Rights to Purchaser.

4.5 SELLER MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, WITH REGARD TO THE DISAGGREGATED SOFTWARE, OBJECT CODE, SOURCE CODE AND PURCHASER'S RIGHTS. SELLER EXPLICITLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR A PARTICULAR PURPOSE. FURTHER, SELLER EXPRESSLY DOES NOT WARRANT THAT THE DISAGGREGATED SOFTWARE, OBJECT CODE, AND SOURCE CODE WILL BE ERROR FREE, WILL OPERATE WITHOUT INTERRUPTION OR WILL BE COMPATIBLE WITH ANY HARDWARE OR SOFTWARE OF THE PURCHASER.

V. LIMITED LIABILITY.

IN THE EVENT ANY PART OR ALL OF SECTION IV ABOVE IS UNENFORCEABLE OR NOT ENFORCED, THEN THE PARTIES AGREE THAT SELLER SHALL NOT BE LIABLE FOR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE AND THAT THE MAXIMUM LIABILITY OF THE SELLER TO THE PURCHASER UNDER ANY LEGAL THEORY, CLAIM, OR CAUSE OF ACTION SHALL BE THE PURCHASE PRICE.

VI. INDEMNITY.

Purchaser agrees to indemnify, defend and hold Seller harmless from any claim brought by any person or entity to whom Purchaser sells, gives, provides, licenses or otherwise distributes the Disaggregated Software for the failure of or defects in the Disaggregated Software.
