

The New Legal Regime, Bad Software, and a Place for Certification

Cem Kaner J.D., Ph.D., ASQ-CQE
Florida Institute of Technology
ISACC Conference, September 2000

Overview

Bad Software

- We've all heard the stories of big-ticket losses due to bad software. I'll skip them and look at routine problems in routine use.
- For more extensive discussion of these "routine" problems, for references, and for quality-related negotiating tips in general, see my book, *Bad Software* (Cem Kaner, David Pels, Wiley, 1998).

Bad Software

- **Software publishers ship products that have plenty of known defects.**
 - **We use cost/benefit analyses to determine which bugs to fix.**
 - **Software test groups pride themselves on the low rate of “surprises” in the field. (A surprise is a bug that the customer finds that had not already been found in testing.) For example, the former Director of Software Testing at Microsoft reported that over a 2-year period, typical products yielded only 2 surprises.**

Bad Software

- The Canadian government recently completed a study of the claims made on the packaging of consumer software: *Incorrect (and “potentially false or misleading”) claims were made by 65% of all the software titles tested.*

Study by Industry Canada’s Competition Bureau. For the full study, go to <http://strategis.ic.gc.ca/FBP> and search for “software”.

- Computer-related complaints made Better Business Bureau’s top 10 for 1995, even higher than used car dealers. We did worse in 1996.

(The BBB’s data for 1997 merged computing with consumer electronics, making comparisons with the 1995 and 1996 data difficult. The combined totals yield higher ranks (more complaints), of course.)

Dissatisfaction Costs Publishers Money

- 1996--200 million calls to tech support.
- The industry spends about \$25 per call.
- Software companies spend about \$3 per minute providing support for PC-based products, and \$5 per minute (or more) for UNIX and mainframe products.
- In companies that have pushed many complainers to the internet, handling the issues raised by live calls cost as much as \$150 to \$400 per incident (averages reported at a 1999 Support Services Conference). (Additional recent data in *Software Engineering & UCITA*, which comes with these notes.)
- Customer complaints have skyrocketed. Over 7 years, ratio of support to total employees in computer-related companies went from 1 in 12 to 1 in 6.
- For references and additional data, see Kaner & Pels, *Bad Software: What To Do When Software Fails*, Wiley, 1998.

Customers Have Legitimate Problems

- In those 200 million calls for support, software customers spent over 3 billion minutes on hold.
- This is tip of the iceberg because most American customers don't complain.
- Cross-industry study: Complaining software customers left on hold for longer than any other industry studied, even airlines and gov't offices.
- At peak times, 85% of calls into tech support get busy signals.
- 58% of support staff get less than 1 week of training before independently handling phone calls.
- Complaints involving software / hardware from more than one vendor take 3 to 18 times as long to resolve.
- Business' cost of ownership of a PC is often estimated at \$8000 to \$11,000 per year.

Sophisticated Customers Have Trouble Too

- At a technical support conference, Albert Stark laid out problems that software support staff have when they buy and install problem management systems. Support staff are an interesting example, because they're so talented at making things work.
- Stark's observations:
 - "The system will not do everything promised."
 - "System functionality is typically overstated."
 - "You'll need to purchase additional modules to get the functionality you need."
 - "Features you need are scheduled for a future release."
 - "The out-of-box reality is less than expected."
 - "You'll need to purchase additional hardware."
 - "The software will be more complex than it appeared during the sales cycle."
 - "System customization will not go smoothly" even though "Vendors can make customization look easy."
- In another session at the same conference, over half a large room of publishers' technical support staff said they would trade in their problem management system if they could.

Publishers Often Ignore Long-Term Cost Analysis

Customer dissatisfaction with quality significantly reduces a company's sales, but several (in my experience, most) companies ignore the dissatisfaction-associated revenue risks because they don't know how to estimate their magnitude.

The degree to which people underestimate long-term effects is illustrated by the following example.

- In the early 1990's, Microsoft spent \$500,000,000 bringing its customer support from blecch to world class. But customer perceptions still rank MS near average as a support provider. Therefore, there might not be an obvious immediate payoff in sales volume. Result--a leading newsletter (read by software senior execs) concluded,

“Despite lots of wishful thinking to the contrary, spending money to upgrade a company's service reputation remains a lousy investment.”

Managing and consulting in Sili Valley during this period, my sense was that MS Office took over its market partially because competitors committed customer satisfaction suicide.

Litigation Over Bad Quality

The essence of quality-related litigation is a customer seeking to transfer losses caused by a defective product back to the company that made the defect or sold it.

The Economics of Quality: Quality/Cost Analysis

- **The Cost of Quality is the total amount the company spends to achieve and cope with the quality of its product.**
- **This includes the company's investments in improving quality, and its expenses arising from inadequate quality.**
- **The primary goal of quality engineering is often described as minimization of quality-related costs.**

Basic Quality Engineering: Quality-Related Costs

- **Prevention**

Cost of preventing software errors, documentation errors, and any other sources of customer dissatisfaction

- **Appraisal**

Costs of looking for defects (all types of inspection and testing).

- **Internal Failure**

Costs of coping with errors discovered during development.

- **External Failure**

Costs of coping with errors discovered, typically by your customers, after the product is released.

- **Total Cost of Quality = Prevention + Appraisal + Internal Failure + External Failure costs.**

Categorizing Quality Costs

<i>Prevention</i>	<i>Appraisal</i>
<ul style="list-style-type: none"> • Staff training • Requirements analysis • Early prototyping • Fault-tolerant design • Defensive programming • Usability analysis • Clear specification • Accurate internal documentation • Pre-purchase evaluation of the reliability of development tools 	<ul style="list-style-type: none"> • Design review • Code inspection • Glass box testing • Black box testing • Training testers • Beta testing • Test automation • Usability testing • Pre-release out-of-box testing by customer service staff
<i>Internal Failure</i>	<i>External Failure</i>
<ul style="list-style-type: none"> • Bug fixes • Regression testing • Wasted in-house user time • Wasted tester time • Wasted writer time • Wasted marketer time • Wasted advertisements • Direct cost of late shipment • Opportunity cost of late shipment 	<ul style="list-style-type: none"> • Technical support calls • Answer books (for Support) • Investigating complaints • Refunds and recalls • Interim bug fix releases • Shipping updated product • Supporting multiple versions in the field • PR to soften bad reviews • Lost sales • Lost customer goodwill • Reseller discounts to keep them selling the product • Warranty, liability costs

Risks of Quality Cost Analysis

- It is too easy to focus on easy-to-measure failure costs, such as Technical Support costs. To deal with these costs, many companies sell support time and turn their bugs into a profit center.
- Lost sales from repeat-potential customers probably cost much more than tech support costs but companies often ignore these issues because
 - They are hard to measure.
 - Customers sink a heavy investment in some products and face huge transition costs if they choose to switch to a replacement.
 - The competition in many software markets is quit thin.

Risks of Quality Cost Analysis

Quality/Cost analysis teaches the company to focus on its own costs, to minimize the sum of quality-related costs that are paid by the company.

This analysis ignores *externalized* failure costs -- the costs absorbed by the customer.

(See my paper, Quality / Cost Analysis: Benefits & Risks, at www.kaner.com.)

What about the quality-related costs that are paid by the customer?

Do You Remember the Pinto?

These were the estimates at Ford:

- External Failure Costs = \$49.5 million
 - 180 burn deaths \$200,000 each
 - 180 serious burn injuries \$67,000 each
 - 2100 burned vehicles \$700 each

- Total Costs to Repair = \$137 million
 - \$11 per vehicle

External failure costs are cheaper than repair, therefore ship it. Right?

This is classical quality/cost analysis, so what's the problem?

Customers' External Failure Costs are Important

<i>Seller: external costs</i>	<i>Customer: failure costs</i>
<i>These are the types of costs absorbed by the seller that releases a defective product.</i>	<i>These are the types of costs absorbed by the customer who buys a defective product.</i>
<ul style="list-style-type: none"> • Technical support calls • Preparing answer books • Investigating complaints • Refunds and recalls • Interim bug fix releases • Shipping updated product • Supporting multiple versions in the field • PR to soften harsh reviews • Lost sales • Lost customer goodwill • Reseller discounts to keep them selling the product • Warranty, liability costs • Gov't investigations 	<ul style="list-style-type: none"> • Wasted time • Lost data • Lost business • Embarrassment • Frustrated employees quit • Demos or presentations to potential customers fail because of the software • Failure during tasks that can only be done once • Cost of replacing product • Reconfiguring the system • Cost of recovery software • Cost of tech support • Injury / death

Summary on Quality Costs

- Software developers and publishers make cost/benefit tradeoffs in determining how they design reliability into their software and how carefully they test and fix it.
- Those tradeoffs tend to focus on the vendor's costs and not on the customer's costs.
- Changes in the legal structure that reduce vendor liability for defects will make it cheaper for the vendor to provide lower quality products or services.

Legal Bases for Lawsuits over Bad Software

- Intentional torts
- Misrepresentation
 - Fraud
 - Negligent misrepresentation
 - Deceptive practices
 - Racketeering
- Malpractice (professional negligence)
- Negligence resulting in damage or injury
- Breach of contract
 - Service contracts vs. Product contracts
 - Warranties
 - Implied warranties
 - Remedies
 - Modification

Intentional Torts

Intentional torts involve unlawful interference with, or harm to, a person or her property, reputation, privacy, or business relations. People who commit intentional torts can be sued for full compensatory and punitive damages.

- **Compensatory damages**: pay back what the victim lost
- **Punitive damages**: pay additional money, beyond compensatory damages, to penalize the tortfeasor for outrageous conduct.

Examples:

- **Assault and Battery**: Actions that cause bodily harm or that make victim reasonably fear immediate bodily harm.
- **Conversion**: Civil law analog to theft. Such as taking victim's property, destroying it, or otherwise treating it as if victim had no property rights in it.
- **Computer tampering**: Cracking, distributing viruses, etc. Unlawful access to the computer of another, or introduction of software that causes damage to the computer of another.
- **Libel**: Publication of false statements about victim that humiliate or cause harm to reputation.
- **Fraud and Misrepresentation**: False statements for profit.

Computer Tampering, Conversion

In *Clayton X-Ray Co. v. Professional Systems Corp.* (WD 43583, Mo. Ct. App., W.D. 8/6/91; 9 Computer Law 38). PSC introduced a time bomb b/c Clayton had not fully paid for the software. The bomb shut down the system. PSC refused to turn the system back on until Clayton paid for it. Jury awarded punitive and compensatory damages.

This is not a case of time bomb in the initial product. This is a time bomb introduced after the fact, by intrusion into plaintiff's computer.

Cases like this have made many vendors cautious about using "self-help" – shutting down customers' software or systems without a court order.

Misrepresentation

- False representation by the seller
- of a material (important) fact
- that the plaintiff justifiably relies on
- and as a result, the plaintiff is damaged.
- Misrepresentation can be:
 - Innocent
 - Negligent
 - Fraudulent
- A misrepresentation is fraudulent if the maker
 - knows or believes that the matter is not as he represents it to be, or
 - does not have the confidence in the accuracy of his representation that he states or implies, or
 - knows that he does not have the basis for his representation that he states or implies
- All states allow you to sue for fraud (and recover punitive damages). A few states allow suits for non-fraudulent misrepresentation (punitives probably unavailable).

Misrepresentation

Negligent misrepresentation

- The *duty* is to exercise the care or competence of a reasonable person who is communicating information.
- Not all misrepresenters will be held liable. Many states require a special relationship between the victim and misrepresenter, such as a position of trust.
- States vary in the degree to which they allow a negligent misrepresentation suit, in the face of an integration clause and no misrepresentation in the body of the contract.

Post-sale misrepresentation

- Post-sale fraud is actionable if it causes a person to forego or refrain from asserting an existing right or to change position in some other way.
- Several support staff are trained to deny knowledge of known bugs and some are told that they are supposed to lie if necessary to keep a customer from returning a product.
- *Ritchie Enterprises v. Honeywell Bull, Inc.* (1990) is a classic case in which the contract successfully disclaimed warranties, limited damages, etc., but the plaintiff was allowed to sue for post-sale misrepresentations (that problems would be or were being fixed).

Deceptive Practices

Uniform Deceptive Trade Practices Act:

A person engages in deceptive trade practices when s/he represents that goods or services have sponsorship, approval, characteristics, ingredients, uses, benefits, or quantities that they do not have.

California Civil Code 1770:

The following unfair methods of competition and unfair or deceptive acts or practices undertaken by any person in a transaction intended to result or which results in the sale or lease of goods or services to any consumer are unlawful:

- (a) Passing off goods or services as those of another.
- (b) Misrepresenting the source, sponsorship, approval, or certification of goods or services.
- (c) Misrepresenting the affiliation, connection, or association with, or certification by, another.
- (d) Using deceptive representations or designations of geographic origin in connection with goods or services.
- (e) Representing that goods or services have sponsorship, approval, characteristics, ingredients, uses, benefits, or quantities which they do not have or that a person has a sponsorship, approval, status, affiliation, or connection which he or she does not have.

Deceptive Practices

California Civil Code 1770:

- (f) Representing that goods are original or new if they have deteriorated unreasonably or are altered, reconditioned, reclaimed, used or secondhand.
- (g) Representing that goods or services are of a particular standard, quality, or grade, or that goods are of a particular style or model, if they are of another.
- (h) Disparaging the goods, services, or business of another by false or misleading representation of fact.

FTC Actions

Go to www.ftc.gov for copies of complaints and settlement agreements involving such companies as Apple, Dell, Gateway 2000, Iomega, and others. Issues include:

- Failure to provide tech support for life after promising to do so
- Failure to disclose key terms of the contract
- Deceptive practices of various kinds

Malpractice

Malpractice is the failure to exercise the skill and knowledge normally possessed by members of a profession or trade.

Historically, courts have not been willing to allow suits for software developer malpractice, but the situation might be changing:

- The wave of licensing of software engineers (currently in Texas, British Columbia and Ontario) will support malpractice claims against individual engineers.
- Y2K-related arbitrations apparently resulted in several successful claims against vendors' errors and omissions policies.
- Software services in conjunction with other professional services (such as accounting) might give rise to a malpractice suit in that other professional domain.

Negligence Resulting in Damage or Injury

- A vendor's products must not create an unreasonable risk of injury or property damage.
- Cost-of-Quality analysis balances seller's prevention, appraisal and internal failure costs (including cost to repair) against the seller's external failure cost.
- Negligence analysis balances seller's prevention, appraisal, and internal failure costs (*especially costs associated with repair*) against society's external failure cost.

Contracts vs Negligence

Contracts

- Law of quality
- Duty is to give the customer what s/he paid for.
- Likely types of suits:
 - corrupts or loses its own data
 - doesn't work; never delivered
 - erroneous reports
 - bugs that waste time or make the program hard to use
 - compatibility features don't work
 - cost-reduction promises aren't realized

Negligence

- Law of safety
- Duty is to make products that are not unreasonably unsafe.
- Likely types of suits:
 - corrupts or loses data obtained from some other program
 - damages connected peripherals
 - injures the user
 - injures customer who follows its directions
 - embedded software causes accidents
 - UI design causes accidents

Contract Fundamentals

- A contract is an *agreement* between two or more people (or companies) that creates obligations to do or to provide particular things.
 - In many cases, there is no agreement-creation process. Instead, we talk of the *voluntary assumption of an obligation* as the basis of the contract.
- A software contract can involve *goods* (such as a program bought at a store) or *services* (such as custom programming), or some mix of the two (such as a program that comes with a maintenance contract).

Traditional Terminology of Contract Law

- Offer
- Counter-offer
- Acceptance
- Consideration
- Battle of the forms
- Warranty
- Modification
 - The pre-existing duty rule
 - Material vs. non-material modifications
- Oral contracts are valid (except under statute of frauds)
- Contracts by conduct are valid
- Remedies

Contracts: Uniform Commercial Code

- Article 2 governs contracts for sale of goods in USA.
- Sale of packaged software has been treated by the courts as a sale of goods.
- Sale of custom software is a sale of services, not covered by the UCC, though several courts have applied the UCC anyway.
- UCC is maintained and updated by
 - National Conference of Commissioners on Uniform State Laws (NCCUSL) a legal drafting organization funded by the 50 US states to write “Uniform” laws. (300 lawyers appointed by states, typically by Governor.)
 - American Law Institute, another non-profit body of senior lawyers (3000 judges, law professors, senior partners, elected to membership by the Institute.)

Uniform Commercial Code

- Article 2 of the Uniform Commercial Code governs sales of goods. Article 2 has been consistently applied to COTS transactions, but much less often to custom service contracts (e.g. custom software, consulting, etc.)
- Interesting features of Article 2
 - Gap fillers, and implied terms
 - Perfect tender rule
 - Implied warranty of merchantability
 - Implied warranty of fitness for a particular purpose
 - Battle of the forms rules:
 - contract by conduct
 - forms as proposals for modification
 - Modification rules: material vs. non-material modifications
 - Some rules apply only to merchants

Uniform Computer Information Transactions Act

- Intended to update the UCC, to handle software more sensibly:
 - Unify treatment of products and services
 - Clarify when a contract is formed
 - Clarify the rules for informing customers of contract terms, including (especially)
 - Warranty terms
 - Remedies
 - Use restrictions
 - Transfer restrictions.
- Governs all contracts involving software and digitally stored information.
- Opt-in clauses can bring in goods sold with software.
- Current draft of UCITA: www.law.upenn.edu/bll/ulc/ulc.htm

Politics of UCITA

- **Typical proponents:**
 - Software publishers
 - Database publishers (West / Lexis / NASDAC)
 - CitiBank
 - Daimler Chrysler
- **Typical opponents**
 - Consumers
 - Insurance companies
 - Librarians
 - Staff members of the Federal Trade Commission
 - 25 Attorneys General
 - American Intellectual Property Law Assoc and IP section of the NY City Bar Association
 - ACM-USA, IEEE-USA, ICCA, CPSR, ASQ, SEI, professional societies and trade groups representing software engineers.

Politics of UCITA

- UCITA was initially an amendment to Article 2. The Article 2 revision committee couldn't reach agreement and split into an Article 2 (law of sales) committee and an Article 2B (law of software licenses) committee.
- The Article 2B project ended when the American Law Institute withdrew, citing concerns that Article 2B was trying to write around federal Copyright law and that it was making fundamental changes in contract law.
- NCCUSL renamed the bill's name to UCITA and went forward on its own. This is the first time that ALI and NCCUSL have split.
- Passed as law in Virginia and Maryland. VA is having hearings this fall (2000) to determine whether amendments are needed.
- Iowa has passed a "bomb shelter" law that invalidates contract clauses that say that UCITA governs contracts.
- Well-funded lobbying effort, expected to reach 20 legislatures this fall.

ALI's Withdrawal from UCC 2B

- ALI passed resolutions in May 1997 and 1998 calling for fundamental revision of Article 2B. It withdrew from Article 2B in 1999. This is from the supporting memo to the May 1998 ALI resolution (Braucher and Linzer):

“The Draft reflects a persistent bias in favor of those who draft standard forms, most commonly licensors. It would validate practices that involve post-purchase presentation of terms in both business and consumer transactions (using "shrink-wrap" and "clickwrap"), undermining the development of competition in contingent terms, such as warranties and remedies. It would also allow imposition of terms outside the range of reasonable expectations and permit routine contractual restrictions on uses of information traditionally protected by federal intellectual property law. A fundamental change of approach is needed.”

What Does UCITA Do?

- My focus today is on quality-related issues.
 - There are big intellectual property law problems too, which many people consider more serious than the quality issues.
 - For discussion of the IP issues, and for a few hundred footnotes that back up my claims in these slides, see my paper, *Software Engineering & UCITA*, in the seminar notes.

What Does UCITA Do?

- Validates almost all of the terms of shrink-wrap / click-wrap contracts.
- These are still controversial and courts are split on the enforceability of their harsher terms. For example, over the last few months:
 - Washington state Supreme Court (*M.A. Mortenson Co. v. Timberline Software*) approved a click-wrap disclaimer of warranties and remedy limitation, barring the plaintiff from recovering significant losses caused by a defect that was apparently known by the vendor but not disclosed to the customer.
 - Federal court in Kansas (*Klocek v. Gateway, Inc.*) just rejected a shrink-wrapped arbitration clause, ruling that this was a proposed modification to the contract.

What Does UCITA Do?

My belief, after thorough research over many years, is that the new wave (UCITA and the court cases influenced by UCITA) represents a radical and fundamental change in contract law.

- Terms are unavailable to customers before the sale (no duty to supply them even on request).
- Terms can be called “conspicuous” even though they are undiscoverable before the sale.
- Material terms will be enforced even though they were unavailable pre-sale and they conflict with the law’s default rules or with normal expectations.

Along the same lines, Geoffrey C. Hazard, Jr., the Chair of the Permanent Editorial Board of the Uniform Commercial Code, spoke at a meeting of the UCC Article 2B Drafting Committee (St. Louis, May, 1998). He said that the process of presenting nonnegotiable, post-sale notifications of the seller's terms that come with software is far from any notion of a voluntary assumption of obligations and from traditional notions of contracting in general.

Examples of Problematic Contract Terms

Here are some of the terms that UCITA will probably make enforceable. Many of these would have been enforceable in a signed, negotiated contract. UCITA applies them equally in click-wraps.

- The terms need not be available to the customer until after she has paid her money and started to install it. Public competition over support and warranty terms will be almost impossible because the terms aren't public.
- Implied warranties disclaimed.
- Product demonstrations are less likely to create express warranties than under Article 2.
- User manual is probably less likely to create express warranties than under Article 2.
- Contract is noncancelable, or cancelable only on payment of a substantial penalty. Contract can't be cancelled even if the vendor materially breaches the contract.

Examples of Problematic Contract Terms

- Remedies limited to a partial refund. No incidental or consequential damages for breach.
- UCITA explicitly eliminates (see comment 6 to section 803) the Article 2 provision for a minimum adequate remedy.
- Eliminates the doctrine of failure of essential purpose of a limited remedy by expressly permitting boilerplate to preserve exclusion of incidental and consequential damages even when an agreed exclusive remedy fails or is unconscionable.
- No duty to warn customer of known defects, and it's OK to charge the customer for support calls arising out of known defects.
- The definition of “material breach” has been changed (fewer defects will be “material.”)

Examples of Problematic Contract Terms

- **Customer not allowed to publish benchmark studies or other reviews without the permission of the vendor.**
- **If a clause in the initial contract (remember this can be a shrink-wrapped clause not available until after the sale) says so, the vendor can change the terms of the contract and the customer may not cancel the contract or otherwise reject the modifications.**
- **Disputes must be settled in arbitration or in court in another state or another country (the vendor selects the forum in the contract).**

Examples of Problematic Contract Terms

- No reverse engineering, not even (within a UCITA-based prohibition) to check for security violations or to create interoperability.
- Vendor can place time bombs in the program.
- Vendor can terminate the license after a “reasonable time”—even a fully paid up license that never specified a term limit at time of sale.
- Vendor can use “self-help” to shut down your use of the software without a court order. There are several restrictions on the vendor’s use of this power, but no risk to the vendor if a 3rd party takes advantage of the security hole and shuts down your system.

Additional UCITA Concerns

- Software products are pulled out of the scope of the Magnuson-Moss Warranty Improvement Act and some other important consumer protection statutes. (Business customers benefit from these rules because they raise the bar for all “consumer” products, no matter who buys them.)
- UCITA provides a few (very few) protections for “mass market” customers. However, mass-market is defined in such a restrictive way that most modestly large companies that buy shrink-wrapped software will not meet the “mass-market” requirements.
- The contracting rules are so favorable for shrink-wrapped products that more expensive software products will be sold under shrink-wrap licenses.

Let's sum this up a different way

In a typical shrink-wrap situation:

- **Limited competitive info**
- **No terms**
- **No warranties**
- **No specs**
- **No duty to disclose known defects**
- **No vendor duty to check its software for viruses**
- **No free support**
- **Nowhere convenient to sue**
- **Virtually no remedies if you win**

These are just the traditional contract issues. The intellectual property rights that UCITA grants to vendors are equally remarkable.

The electronic notification rules are also seriously problematic.

What to Do About UCITA (Politically)

■ Virginia

- The Virginia legislature will consider amendments to UCITA this winter, based on recommendations on the Joint Commission on Technology & Science.**
- Members of the public are invited to speak at the meetings and to introduce specific amendment language. Information about the Advisory Committee, the dates and locations of the upcoming meetings and the procedures to follow for participation is available on the Committee's web site**

<http://jcots.state.va.us/documents/00-01/00UCITA.htm>

What to Do About UCITA (Politically)

■ Nationally

- The 4CITE coalition (“For a Competitive Information & Technology Economy”) is the leading opposition group.
- They’re operating on a shoestring budget and need money NOW.
- Additionally, they want more businesses to publicly list their support of 4CITE or their opposition to UCITA in some other way.
- www.4cite.org
- Contact "Alan Kitey" <AKitey@hrrc.org>

What to do About UCITA (Negotiated Terms)

- I've taken several of my notes from the following paper. Elaine was active throughout the UCITA meetings and has a thorough knowledge of the bill's traps. I may be able to distribute her paper at the conference. If not, you might ask her for it yourself:

**Protecting Your Business Client Against the Pitfalls
of UCITA: Practice Tips for Licensees' Counsel**

By Elaine McDonald, J.D.

Principal Financial Group

(515) 247-5675

mcdonald.elaine@principal.com

What to do About UCITA

- Review user documentation prior to accepting a license (this will support your efforts to treat the docs as an express warranty regarding any statements of fact).
- Ask for disclosure of all known defects.
- Send confirmatory memos that describe the specific promises and statements made during sales presentations and discussions. These may not prove breach of contract (because of the contract's integration clause) but they may prove fraud or misrepresentation.
- Ask for a specification.
- Purchase by soliciting responses to an RFP and specify (either by memo or by a clause in the contract) that the contract incorporates the vendor's statements in the RFP.

What to do About UCITA

- **Ask for guarantees. Examples of things that you can consider asking for:**
 - **Certification by a third party test lab that the product was tested to a commercially reasonable degree, that the lab has reviewed the vendor's known defects, and the lab is not aware of serious, undisclosed problems in the product.**
 - **Certification of product quality in some other way.**
 - **Certification that the product was developed according to a specified process (see the new UL standard for embedded software, or SEI CMM, etc.)**

What to Do About UCITA

- In your contract negotiations, ask for support, such as:
 - Answer / escalate / resolve within X time
 - Free support for X time
 - Free support for defects. Here is my current definition of defect:

“In a product that is designed and developed primarily by the vendor or by a supplier to the vendor, a "defect" is "a failure to conform to the reasonable expectations of a reasonably well informed customer, based on information supplied to the customer by the vendor and on other published information, available to the public at the time of the transaction, that would set the expectations of a typical customer for a product of this type. In the event of a conflict, information supplied to the customer by the vendor is the more authoritative source for expectations of the product.”

What to Do About UCITA

- In the event of a dispute, UCITA will favor the vendor in breach-of-contract lawsuits, so ...
 - During the acquisitions and support process, keep records of the vendor's statements (send confirming memos, etc.). The goal is to build a record to support a lawsuit for fraud, misrepresentation, or deceptive trade practice.
 - Look for ways to establish a standard of care or a standard of competence for custom work. The goal is to create a basis for a suit for malpractice or service-provider negligence.
- Ask the vendor about litigation and arbitration history associated with this product or products of this kind. (You might learn something useful or you might get bland but false reassurances that there were no problems, which could be useful for proving misrepresentation later.)

Custom Software Development: Dispute Resolution

- The typical contract considers the possibility of a complete breakdown of the agreement. It lays out procedures for dealing with these types of disputes.
- Consider adding dispute resolution procedures that you can follow mid-project, that are designed to save the project rather than to help you fight over the failure. (For more on this, see <http://www.kaner.com/contrac2.htm>.)

Finally, if you get into a lawsuit

- **Choose and supervise your legal team with care**
 - Many plaintiffs counsel are clueless about computer law
 - Many plaintiffs counsel (especially on contingency) do detailed preparation at the last minute, in contrast with defense counsel (paid by the hour) who often prepare more thoroughly much earlier. Unprepared plaintiffs counsel sometimes concede key points or key issues early without realizing what they're throwing away.
 - Many commercial lawyers don't realize the extent to which the law is changing and are stunned by the degree to which computer products law might be treated by a court in their state differently from other Article 2.

Finally, if you get into a lawsuit

Choose and supervise your legal team with care

- **Your commercial counsel might be too sympathetic to the vendor's side and unable to aggressively pursue your interests.**
- **Look carefully at your counsel's choice of experts for pre-trial consulting and testimony. A technical lawsuit is often significantly developed through the research and testimony of the experts.**
- **If your case goes to appeal, press your counsel to arrange for amicus briefs and to provide potential amici with plenty of notice and (within ethical limits) support.**