CEM KANER, J.D., Ph.D.
Law Office of Cem Kaner kaner@kaner.com
P.O. Box 1200 408-244-7000 (Voice)
Santa Clara, CA 95052 www.badsoftware.com 408-244-2181 (Fax)

UCITA: Problems for Buyers and Developers

Notes for the talk at Roger Williams University, April 22, 1999

David has talked about UCITA in a way that suggests that "the software industry" is united behind UCITA. Everyone on the drafting committee speaks this way. So do most supporters of UCITA and most *lawyers* who are critics of UCITA. But nothing could be further from the truth.

<SLIDE 2>

- The Association for Computing Machinery is to software developers about what the American Bar Association is to lawyers. And the Association for Computing Machinery has asked that the 2B/UCITA project be terminated.

<SLIDE 3>

- The Institute for Electrical and Electronic Engineers is another large association that includes many software developers. The IEEE has written NCCUSL twice with strong criticisms of 2B, and indicated that if its objections are not dealt with, it will seriously consider recommending termination in the future.
- The Independent Computer Consultants Association is the primary trade group for individual software development consultants and small development shops. When I started working on Article 2B, back in February of 1996, one of the frequently repeated claims by the drafters was that 2B was designed to benefit small developers as well as large publishers. That claim was made time and time again until the Independent Computer Consultants Association wrote NCCUSL asking that the 2B project be terminated.

<SLIDE 4>

- Article 2B also covers the work products of technical and freelance writers, at least to the extent that their writings are published or posted electronically -- as is the case for more and more of these works. The primary association of technical writers is the Society for Technical Communication. I'm a Senior Member of that Society. We would probably be in opposition to 2B, but for internal reasons, the Society avoids taking political stands. The National Writers Union might be the next largest group representing technical writers even though it is much smaller, representing about 5,000 tech writers, journalists, and book authors. The National Writers Union has written to NCCUSL asking that the 2B project be terminated.

<SLIDE 5>

- The software-test-discuss forum is the primary internet e-mail discussion group for software quality control engineers, and the closest thing we have to a membership organization for software testers. After extensive discussion, and with no opposition from the 400-plus members of the list, the group wrote to NCCUSL asking that 2B be terminated.
- I am not aware of any professional society or other group representing individual software developers that has expressed support of Article 2B or UCITA.

Software publishers' lawyers and executives, the very large software consulting firms and the mass-market computer makers who will be able to bring their computers within the scope of UCITA certainly do seem to support the bill. But the people who design and develop these products do not support UCITA. The industry is far from being united behind UCITA.

<SLIDE 6>

What I hope to achieve today is to give you a look at UCITA through the eyes of a software quality engineer. Laws have effects. UCITA will have big effects on the practice of software engineering.

Let's start out with one of the fundamental premises of quality engineering, as laid out by the American Society for Quality. That is that a primary function of quality engineering is to minimize the total quality cost of a product. We calculate that cost as the sum of four categories of costs, prevention, appraisal, internal failure and external failure costs. Prevention costs include money we spend money to prevent defects. Appraisal costs cover all methods of testing or looking for errors. Failure costs include all the money that we spend to deal with defects in the product. Internal failure costs are triggered before the sale, for example the cost of fixing a defect before shipping the product. External failure costs involve the costs of coping with the consequences of shipping defects to customers. The goal of the quality engineer is to help the company minimize the sum of these costs.

<SLIDE 7>

Here are examples of the types of quality costs. External failure costs include lawsuits, warranty claims, refunds, lost sales -- these costs are huge. The size of these costs serves as an incentive to improve product quality. An ounce of prevention can be worth many pounds of external failure costs.

<SLIDE 8>

There's a problem with quality cost analysis. It focuses our attention on external failure costs -- the costs that accrue to the ***company*** that ships or sells a defective product. What we don't see are the costs to the ***customer*** of that defect. The costs to the customer are ***externalized*** costs. Externalized costs are often taken into account in Japanese engineering, but they are often ignored in America by manufacturers and by commercial lawyers. Somehow it seems that if you can externalize a cost, you can pretend that it has miraculously gone away. That's the kind of thinking that gave us the Pinto, and many other fine products that you've read about in your torts casebooks.

UCITA takes this problem to its worst extreme. In its drive to shield software publishers from liability, UCITA cuts down a remarkably wide range of external failure cost risks. In doing so, UCITA makes it easy and tempting for publishers to reduce their products' quality, helping them to ship worse products faster. This drives the externalized costs, customers' costs, through the roof.

Let me suggest that American customers won't stand for this forever. As with so many other industries, when customer abuse gets severe enough, we'll see waves of regulation. And abandonment of American manufacturers in favor of foreign competitors.

I think that we're getting close to that threshold of abuse today. I think that UCITA will help push the industry over that threshold. I don't understand why industry after industry has to learn, the hardest way, that what you find when you cross that threshold is a cliff.

<SLIDE 9>

Let's take a look at this mess.

Let's start with the publisher claim that they need relief from liability because it is impossible to make perfect software. I have a lot of sympathy for this argument. If you ever manage software development, as I have, you will run into the situation where you release a product believing that it is in good shape and then have customers call you to report a problem that you never dreamed was in the product. This happens to every publisher. The idea of facing high liability for this is terrifying.

As the author of the best selling text in the field of software quality control, I've been teaching for years that we don't know how to make perfect software or how to test it completely.

But this argument only stretches so far.

We might not want to hold publishers liable for defects that they didn't know about and that they had no reasonable chance to find. But what about defects that they did know about, that they chose not to fix, and that they chose not even to disclose to their customers. A publisher who won't even give his customers a chance to mitigate their losses hardly deserves a break for those known, hidden defects. But UCITA makes no distinction. Publishers get a free ride (or at least a very cheap ride) for all defects, known and unknown.

<slide 10>

There is a mythology about publishers' inability to cost effectively manage their quality-related risks. Let's burst some of those myths:

We've already looked at the problem of known defects.

The next issue is the claim that most customer complaints aren't really about defects. Customer