# The Balance Of Software, Ethics, Professionalism

Scott Barber

Counter to a common presumption, neither software testing nor any software development role, for that matter, is legally considered a profession in the majority of states in the U.S. Occupations related to software development are becoming collectively recognized as a profession known as software engineering—and while some in the field applaud this, others adamantly oppose it.

What I find most fascinating, however, is that few people currently in occupations related to software development—people who believe they're part of a profession—are completely aware of the implications of that belief.

I'm not criticizing these individuals. I'm mentioning this simply because that belief, coupled with a lack of awareness of its implications, is actually detrimental to the software development industry as a whole—it can lead people without significant knowledge of the software industry to assume that some minimum degree of skill and knowledge is required by law to carry a software development–related title (not to mention creating uncomfortable situations for individuals who find themselves caught unaware).

## A Pro Is as a Pro Does

Before I go any further, I want to make something extremely clear. When I say that software development–related occupations aren't professions, I'm not implying that the individuals in these occupations don't conduct themselves in a professional manner. Nor am I implying that those who do act professionally don't deserve the respect afforded to established, recognized professionals. Quite the contrary: Many members of the industry have earned—through their education, expertise, accomplishments, ethics and community standing—the right to the same respect as those belonging an established profession. The point here is to distinguish between observing how a person conducts him or herself individually and making assumptions about an individual based on a belief that he or she is a part of a recognized profession. Allow me to explain.

According to the State of Pennsylvania Associations Code (Title 15, Chapter 1), which I'm referencing as an example because it's short, clear and generally representative, a *profession*:

> Includes the performance of any type of personal service to the public that requires as a condition precedent to the performance of the service the obtaining of a license or admission to practice or other legal authorization from the Supreme Court of Pennsylvania or a licensing board or commission under the Bureau of Professional and Occupational Affairs in the Department of State.

Wikipedia discusses the concept of professions colloquially, as follows:

> In modern usage, professions tend to have certain qualities in common. A profession is always held by a person, and it is generally that person's way of generating income. Membership in the profession is usually restricted and regulated by a professional association. For example, lawyers regulate themselves through a bar association and restrict membership through licensing and accreditation of law schools. Hence, professions also typically have a great deal of autonomy, setting rules and enforcing discipline themselves. Professions are also generally exclusive, which means that laymen are either legally prohibited from or lack the wherewithal to practice the profession. For example, people are generally prohibited by law from practicing medicine without a license, and would likely be unable to practice well without the acquired skills of a physician. Professions also require rigorous training and schooling beyond a basic college degree. Lastly, because entrance into professions is so competitive, their members typically have above-average mental skills.

In both cases it's clear that occupations related to software development fall short of meeting common criteria that qualify fields as professions. We certainly have no software development association that restricts membership, accredits training programs and enforces discipline on its membership.

Another interesting piece of data that is not widely known: The United States, at least, has no legal precedent that establishes the tort of computer malpractice or of professional negligence in software development or software engineering. Needless to say, abundant legal precedent exists for the tort of malpractice and professional negligence in the recognized professions of law, medicine and engineering, for example.

This makes sense, really. After all, you must demonstrate no minimum skill or knowledge to become a software developer or tester. In fact, you can become a software developer or tester with no training whatsoever. It certainly wouldn't seem reasonable to hold an individual legally liable for negligence in having failed to apply practices that he wasn't required, or often even expected, to know in the first place, would it?

Scott Barber is the CTO at PerfTestPlus. His specialty is context-driven performance testing and analysis for distributed multi-user systems. Contact him at sbarber @perftestplus.com.

## Unfairly Accountable?

What this means is that the company publishing the software (serving as the employer of software developers and testers) is legally responsible for ensuring that software is of appropriate quality—*not* the individuals the company employs to develop or test that software.

This also makes sense, since an individual developer or tester rarely makes the decision to publish the software. On the contrary, in a recognized profession such as law, medicine, accounting or engineering, it's not just the employing organization that is legally responsible for the quality of service delivered by its employees (though the organization may also be responsible); instead, the individual providing the service can be, and often is, held personally responsible for substandard services.

For an example of how that accountability might work, imagine that someone purchased a software product that you had once contributed to (be it as a developer, tester, analyst or some other role). Now imagine that the purchaser found that the software didn't perform according to reasonable expectations in some way.

Further, imagine that after the call to customer support and the root cause analysis, it was determined that the bug/issue/defect was in your code, or in the area of the application that you tested or collected requirements for.

Finally, imagine that the individual who purchased that software was then legally able to sue you, personally, for professional negligence or malpractice. This is exactly what happens when you become part of a recognized profession, and someone decides he's displeased with the quality of your work.

I don't know about you, but the thought that I could be personally sued for not writing code to catch some obscure error condition or for not detecting some bug that appears only on some rare configuration that I didn't get a chance to test based on the schedule and the company's assessment of risk frightens me more than a little.

Worse, what if I *had* detected and reported the issue, but the company had decided to release anyway? Worse still is the fact that, for all practical purposes, it's not possible to completely test software—and even if it were, there's no generally accepted standard that I could reference to demonstrate that I had done my job appropriately.

Since we aren't "professionals," we don't have anything to worry about. The worst thing that can happen to us for doing a really bad job of developing, testing or designing software is that we can get fired. With no legal precedent for computer malpractice or professional negligence, we can—and some do—get away with doing shoddy work with no real consequence.

*In a recognized profession, an individual can be held personally responsible for substandard services.*

## 'Professional Quality'

While I have no interest in being legally liable for the quality of software that I have no final say in releasing, nor am I advocating for a state license to practice software development or testing, the absence of state standards or legal liability is no excuse for doing less than excellent work. I do believe in personal responsibility and doing "professional-quality" work, as I'm sure that most software developers and testers do.

So how do you demonstrate to your employers, clients0 and users that you, as an individual, are personally committed to doing our best work, staying up-to-date with your knowledge and training, and reporting issues and risks honestly and ethically?

You can join an organization or association dedicated to your specialty that requires its members to accept and abide by a standard of ethical behavior. Several such organizations are relevant to software development–related occupations as a whole; for example, ACM (the Association for Computing Machinery) has a special-interest group for software engineering, and IEEE (the Institute of Electrical and Electronics Engineers) has a computing society, but until recently there was no nonprofit, professional-class organization dedicated specifically to software testing.

Now there is. The nonprofit Association for Software Testing (www.associationforsoftwaretesting.org) is dedicated to advancing the understanding and practice of software testing. AST serves academics, students and software development practitioners. In September of 2006, AST adopted the ACM Code of Ethics as a series of principles to guide and govern practice among its membership.

In the spirit of full disclosure, I'll tell you that I am a founding member of this organization and the board that voted to adopt the code of ethics, but that's not why I'm sharing this with you. I'm telling you this because I believe that many individuals who mistakenly refer to themselves as software development or software testing professionals do so in an attempt to express that they take their careers seriously, strive for continual improvement, are committed to quality software, and contribute (or at least try to contribute) to the advancement of their field.

It is my opinion that membership in one of these organizations serves to make those claims about an individual and his or her work. As a member in good standing of AST, ACM or IEEE, you're explicitly stating that you've personally accepted these responsibilities. In my view, that's a powerful testament to your sense of personal responsibility for the quality and ethics of your work. And that matters more to me than whether or not I could be held legally liable.

If you're not already a member, I encourage you to look into ACM, AST and IEEE, and to give some serious consideration to joining one (or more) of these organizations that fits best with your area of specialization.