from the editor

Editor in Chief: Hakan Erdogmus  National Research Council Canada  hakan.erdogmus@computer.org

The Seven Traits of Superprofessionals

Hakan Erdogmus

In their IEEE Software guest editors’ introduction titled “Professional Software Development—Fact or Fiction?” (Nov./Dec. 1999, pp. 13–18), Steve McConnell and Leonard Tripp point out that “professionals see themselves as part of a community of like-minded individuals who put their professional standards above their individual self-interest or their employer’s self-interest.” Using this implicit definition of a professional as a jumping-off point, I want to focus on a subset of conscientious professionals whose personal standards demand certain behaviors. I call them superprofessionals, and the mode of conduct to which they adhere superprofessionalism.

I could characterize this ensemble of behaviors as “high integrity” if “high integrity” were an absolute concept, but it’s not—so I’ll avoid this claim. You might argue that the resulting standard is naïve in its intensity and realism. Although I cannot delineate superprofessionalism compactly and definitively, I know it exists. The best I can do is to list high-level markers that candidates display.

Why the Distinction?
I believe this group is small in software development practice. It would be too easy to virtuously proclaim the underlying behaviors to be the normal rituals that all should observe. But that’s easier said than done. Such a sweeping proclamation would not come close to approaching the reality of common practice. Lots of things get done and more than plenty of software gets developed without everybody conforming to the same super-high standards.

I don’t intend to characterize what misconduct or unethical behavior is. For those interested, Brian Berenbach’s and Manfred Broy’s Computer article “Professionalism and Ethical Dilemmas in Software Engineering” (Jan. 2009, pp. 74–80) does an excellent job in this regard by describing common behavioral patterns that lead to professional misconduct with potential ethical and legal implications. My purpose is different here.

Neither do I offer the concept of a superprofessional as a veiled criticism of the state of software practice. I merely offer it as an observation, as a hopefully useful way of distinguishing a small crowd of practitioners who don’t necessarily stand out in other obvious ways. I’m intent on decoupling superprofessionalism not only from ethics, but also from proficiency and competency. Surely there are significant overlaps among these notions beyond the philosophical, but I’ll treat superprofessionalism in its own right rather than as a vehicle for ethical conduct or a substitute for great expertise or superior ability. I will return to these differences later.

What Is It?
So what’s superprofessionalism then? It’s general professional behavior characterized by seven high-level markers.

Focus on Individual Responsibility
An overriding sense of individual responsibility and urgency to deliver promised outcomes is a central marker of superprofessionalism. The individual’s personal professional responsibility to peers, em-
ployer, and clients consistently trumps self-interest. In cases where clients’ and employers’ interests clash, personal responsibility to clients beats the interest of the employer, but without being unfair to the employer. Where group and personal responsibility clash, personal responsibility wins. This sense of responsibility exceeds the person’s normal legal or contractual obligations or those of the person’s group or employer. Focus on individual responsibility leads to behavior that goes beyond the regular call of duty. It also implies duty above enthusiasm: no task, however menial, is beneath the superprofessional when it needs to be done.

Acute Awareness
A highly developed and sophisticated sensitivity to stakeholders’ needs and interests is another hallmark of the superprofessional. However, awareness entails self-awareness as well as externally directed sensitivity. Self-awareness implies sincere humility, knowing yourself, including your strengths and weaknesses, knowing what you know, and knowing that you don’t know everything. It implies confidence in your own abilities, but also recognizing areas and tasks that require more proficiency than you possess and more time than available. It means willingly, not reluctantly, seeking help when help is needed.

Careful observation of the environment and self often makes a person change behavior to adapt to new situations: awareness leads to learning. Cultural sensitivity is another dimension of awareness that relates to the capacity to adapt behavior and actions to the cultural context in which a person practices a profession. Superprofessionals display this versatility by trying to understand the norms of new organizational, social, and geographical groups in which they find themselves and adjusting their conduct to respect those norms as best as they can.

Brutal Honesty
In their IEEE Software article (“Lying on Software Projects,” Nov./Dec. 2008, pp. 90–95), authors Robert L. Glass, Johann Rost, and Matthias S. Mattock reveal the pervasiveness of lying, in particular in activities such as estimation and status reporting. Distortion of facts—whether lying, fibbing, exaggerating, avoidance, or sugarcoating—is apparently so common that it’s almost expected and accepted. This statement should be shocking. Alas, it’s not.

Superprofessionals’ actions are guided by their commitment to facts regardless of the unpleasant repercussions, whether seemingly harmless or not, of concealing, warping, or avoiding the truth. The pressure to diverge from facts is so intense, tempting, frequent, and unpredictable that this trait must unfortunately be rare.

Resilience under Pressure
Sustained commitment to facts is possible if an individual can withstand constant pressure from peers, clients, employers, and sometimes the self. Resistance to pressure is one type of resilience. But such resilience may not be possible without personal sacrifice. Resilience might imply foregoing promotions, incurring financial loss, or even losing one’s job.

The second component of resilience entails not lapsing in one’s professional commitments under changing priorities. Like the first type, the second type of resilience might produce negative personal consequences. But should we expect such selfless conduct from everyone all the time in the name of professionalism? No. So, like commitment to facts, resilience is a rare marker.

Heightened Sense of Fairness
Conflicts of interest are pervasive in software development. In his blog on professional software development, Basel Vandegriend writes about how conflict of interest is unavoidable in software development contracting (www.basvl.com/psd/blog/2006/is-ethical-consulting-possible). Vandegriend advocates appealing to personal professional ethics to overcome clashing interests. On one hand, I admit that certain aspects of resolving a conflict-of-interest situation could be subjective. On the other hand, an intrinsic, universal notion of fairness exists. The superprofessional has a highly developed sense of fairness and applies it diligently when balancing conflicting interests.

A general trait that precedes a clear and objective sense of fairness is the willingness to share credit and recognize others. The superprofessional refuses to accept credit when it’s due, without a display of false modesty. They are instinctively careful about giving credit where credit is due.
Attention to Detail in Perspective
The devil tends to creep into the details in the software profession. But it’s not easy to attend to the details without losing the bigger perspective. The risk of getting lost in the details is real.

Attention to detail is not obsessive perfectionism: it’s recognizing the impact of low-level decisions and subtle technicalities on a project’s success. A software practitioner might face tens of microdecisions per day. Those microdecisions can embody multiple trade-offs among cost, quality, schedule, and business goals when dissected at a sufficient depth. The superprofessional addresses the details while maintaining this bigger perspective.

Pragmatism First
Davy Brion talks about pragmatism’s role in his blog entry on software development ethics (http://davybrion.com/blog/2009/01/ethics-in-software-development-pragmatism-over-dogmatism). Using a compelling example, he stresses that delivering value to clients and users trumps sticking with dogmatic principles of craftsmanship, loyalty to favorite practices, or any overdeveloped sense of pride in work when you’re making hard choices involving cost/benefit trade-offs. Pragmatism is the propensity to balance those difficult trade-offs, if necessary, by compromising on personal convictions and pride in work.

What It Isn’t
In the beginning of this essay, I mentioned my insistence on differentiating among superprofessionalism, proficiency, and ethics. This point warrants further explanation.

Proficiency vs. Superprofessionalism
Proficiency is much more relative than superprofessionalism. We can’t know everything there is to know about our broader fields or be able to perform every task at an expert level. Nor can we occupy all professional roles. That is, we’re more competent at certain tasks and roles, less competent at others, and not at all competent in still others. Superprofessionalism isn’t universal competence. Designers, requirements engineers, software engineers, software architects, business analysts, developers, testers, quality experts, process experts, project managers, technical leads, program managers, and software executives all claim superior proficiency in only a subset of software development activities, but they can all be superprofessionals. Yes, superprofessionals are highly proficient in their core roles, but they aren’t necessarily competent in all things related to their larger profession. Conversely, not all experts or gurus are superprofessionals, nor do they have to be to play leading roles in their respective communities.

Ethics vs. Superprofessionalism
The IEEE Computer Society and the ACM have jointly defined a comprehensive set of guidelines that describe ethical and responsible professional behavior in software engineering practice (see www.computer.org/ethics). The ACM/IEEE-CS code addresses professional conduct under eight general behavioral areas. The code’s articles are open to interpretation, as they should be. They don’t provide explicit traits for recognizing an exceptionally high level of adherence to the advocated behaviors.

Superprofessionals behave ethically not because it’s prescribed by a code of conduct, but because not doing so would violate their personal professional standards. In the February 2009 edition of his e-newsletter addressing ethics in software development (Food for Thought, vol. 6, no. 2, www.swqual.com/newsletter/vol6/no2/vol6no2.html), Steve Rakitin introduces the notion of an “internal code of ethics” for handling situations and decisions that software professionals face every day and for which formal codes of conduct provide little guidance. Although one way of viewing superprofessionalism is as an internal code of ethics, I prefer not to establish a direct equivalence; I’ll leave it to you to decide on the strength of the connection to and implications for ethics. Personal notions of ethics vary widely. Also, highly professional behavior is more than just ethical behavior.

Do They Exist?
It’s unrealistic to expect a large number of individuals to exhibit superprofessionalism continually and resolutely in a competitive, complex industry such as ours without unduly taxing themselves (or breaking into a self-righteous hive). Most highly disciplined people need to let go from time to time and pick their battles carefully to survive. But I’m privileged to know many colleagues with extremely successful careers who display the general markers without apparently burning themselves out. And they’re plainly and honestly self-aware when they diverge from the underlying behaviors, thus bootstrapping acute awareness, a central marker of superprofessionalism.

If organizations want to encourage superprofessionalism, wouldn’t adopting the seven traits as core organizational values be an obvious start?