

Feature Articles

TopCoder @ Work: Professional Ethics

[Archive](#)
[Normal view](#)
[Discuss this article](#)



By **timmac** & rhudson
TopCoder Members

Staccato bursts of furious typing and clicking echo through the halls, turning an otherwise listless Tuesday afternoon into a frenzy of activity. Yet, this mad pulse of action does not signal productivity. To your right, Sandra sends rapid keystrokes to her French boyfriend in an Instant Messenger Window. To your left, Bill deftly maneuvers a fleet of Space Marines in an all-out assault on an alien hive. In front of you, your own monitor betrays you by displaying your personal e-mail account and a window full of source code -- custom-tailored for your night-job freelance client.

For a moment, it occurs to you that you are actually doing work for another company while you're at work. Pangs of guilt and fear slice through you like the first swallows of Mountain Dew in the morning. What tragic series of circumstances changed you, Puritan Programmer, into an Ethically-Challenged Evildoer?

Blame the computer -- by providing near limitless flexibility and unparalleled access to resources, technology has lubricated the downward slide of ethics for both employer and employee. As a worker, you face opportunity for distraction and time-stealing occupation. As a manager, you seek a panacea -- a cure-all predicated on the idea that computers can solve any problem quickly, and without human consideration. People must transcend the temptations of technology in the workplace and develop a shared view of ethics: what is "right," what is "wrong," and what is "fair."

These "three degrees" of ethics are further complicated by restrictions set by company rules, contracts, and laws. All three categories may be violated by the simple use of web-based e-mail during a lunch break. Visiting "unauthorized" sites, including webmail, may be prohibited by company policy. Opening a message attachment may unleash malware on the corporate network. Once compromised, company computers can be coerced into revealing private data or attacking outside systems. Chaotic consequences can ensue from behavior that doesn't seem ethically "wrong."

"Wrong," or unethical acts can't simply be identified by activities that hurt the company or have the potential to cause harm. "Damage" is difficult to determine, and is often gauged by the slippery metric of "lost productivity." To a strictly "by the numbers" manager, surfing the Internet for an hour a day amounts to a productivity "loss" of five hours in a standard work week. Are the five hours truly lost? The surfing employee may be energized to accomplish more after a "head-clearing" period of web browsing.

These conflicts between notions of what is "right" and what is "wrong" are resolved -- often subversively -- by what an employee considers to be "fair."

In this article, we'll highlight some of the more common ethical quandaries you're likely to encounter in your work (in Part II, coming next week, we'll lay out some trickier, "gray-area" situations). Before we go further, though, there are a few basic points to keep in mind:

- Know the company policy. Start with the employee manual, or written documentation, but don't stop there. Often, the way a policy is written and the way it is put into practice are two entirely different things.
- Call it common sense, or your better judgment, or a moral compass, or The Force -- whatever it is, use it. Whether or not we choose to follow it, all of us have that sense of what's right and what's wrong. Listen to it.
- If you aren't sure, ask! Are you worried what your boss might think if he sees you checking your personal e-mail? Unless he's already been clear on the issue, he will probably appreciate your honesty in asking first.

Personal Use of Company Property (Surfing the 'Net at Work)

Survey data is abundant on this issue and the reality is, a majority of people spend at least some portion of their time at work just surfing the net (or checking their e-mail, etc). What's worse is that surveys of employers about what fractions of time might be

considered acceptable are certainly out of alignment with reality. A typical manager might say that, officially, no personal use of the internet, or very limited, is acceptable. Most of the same managers would acknowledge that some amount of such usage is common, and accept it without another thought. (It's not entirely a victimless crime, either. Bandwidth-hogging sites like MySpace and YouTube could eat up serious amounts of network availability.)

This is not an easy situation to figure out – standards vary widely from company to company, and even from manager to manager. A firm knowledge of company policy with a good dose of common sense should give you a good foundation, however. How well one keeps up to date on their work-related tasks likely figures into the equation, as well. Though I might do a double take at the contractor who is always browsing eBay listings for baseball cards, if his work is consistently ahead of schedule, and of top quality, then this may not be an issue. If you're in a position to set the standards, it will be up to you to decide -- is the work getting done the only thing you care about? Or is how people manage their time just as important?

Using Code from the Internet

You found a terrific open-source socket routine on SourceForge and integrated it into your project at work. The only problem is that the SourceForge license doesn't permit commercial use of its routines. Your company now wants to sell a wrapper for the socket routine, but you haven't mentioned that you "borrowed" the code...

Most people wouldn't worry -- common routines and time-saving code snippets may not even be visible in the final product. But are these routines 100% functional? Can you debug "found code" once it's implemented in your company's production system? The "black box" nature of object orientation allows even the most elementary coder to drop in a component without doing much more than providing the input and output. The process can be a mystery, whether or not you have the source code!

One of my co-workers provides the following rule of thumb: "If you don't understand how it works, then don't use it."

Outsourcing your Code

You're on a typical rushed schedule, but you're paid *very* well. In fact, you even have money to spare. Why not spend this surplus cash on a subcontractor from RentaCoder.com? With the extra help, you can bill the hours it would take *you* to finish the gig while paying someone less to do the job while you sleep. If the job is well-defined, specifying the work isn't too tough, and you can make some extra cash.

Assuming you're not trusting the subcontractor with company secrets, what's wrong with this scenario? Is it fair to charge your client full-price for work you're getting on the cheap? Do you have to disclose the source of the work at all, if you provide a working final product?

Working Side Jobs

The non-compete clause in your contract boils down to a pretty basic issue: your employer wants your full energy. Depending on the company, the standard agreement may range anywhere from a very specific "may not work for a direct competitor" to "may not work for any other company in this line of work". Legally speaking, you are bound to whatever you sign. Whether or not it is fair is another matter entirely – odds are, though, you don't want to have to argue that point in court, or lose your job over it.

Why might you want to work side jobs? More experience, extra money, an altruistic desire to help a friend who is just starting out his own business. Whatever the case may be, side jobs can be tricky business. Possibly the most important consideration is how it may impact your day job. Too much extra work will, at some point, risk leading to a burned out state that spreads into your full-time employment. Regardless of whether you tell your employer about your other commitments, odds are they're eventually going to notice.

On the other hand, a moderate amount of side work can also provide exposure to new technologies and opportunities to expand one's skill set, in ways that could have a very positive impact on your day job. While it is typically best to avoid borrowing code from your day job to use on the side, or vice versa, experience is something you can take along and use anywhere.

Job Hunting

You've just pulled another 60-hour week, 20 more than the "40 hour workweek" that guides the rest of the world, and you're ready to quit. Yet, you have neither the time nor the energy to search for jobs outside of "working hours." Is it okay to use working hours to search for a job because your company "owes you" 20 hours for your unpaid overtime?

This is a consideration that applies both to job hunters and to those working side jobs. How do you quantify what counts as work: the amount of time you spend working? The volume of code or paperwork you generate? The percentage of project goals that you have delivered on? Maybe you work longer hours than others, or maybe your work is of higher quality. Maybe the project itself is just not that important. Whatever the reason, don't you deserve at least a little extra consideration?

The answer, of course, is "no," or at least, "not exactly."

It is basic human nature to give yourself the benefit of the doubt (and, accordingly, to judge others a little more harshly than you judge yourself). Keep this in mind, and don't base your decisions too much on what seems fair to you. Instead, take a step back and think about what's fair for everyone – you, your employer, your co-workers and clients, etc. – and you should be able to come up with some standards that are decent, flexible and realistic.

As a personal rule of thumb, I will respond to an occasional e-mail about a side job or job application (or, ahem, a freelance writing assignment) with simple questions that can be answered quickly and concisely, but anything that requires more than a minute or two really needs to wait. For instance, if I get an e-mail requesting a small job to be done, I might respond by saying that "this looks like something I can have done by next week, I'll call you later (on my way home) and discuss it with you." In such a way, you have acknowledged the message, and set a time when you can deal with it that is outside of your dedicated work hours. If you regularly work outside jobs with the same individuals, setting clear expectations about these kinds of things upfront is typically well respected, and avoids potential problems later on down the road.

Work Quality

Your company produces two versions of an application, one that works online, and the other that works as a desktop application distributed on CD. One single source of raw data powers both versions, but the production of each application has a unique conversion process that compiles that data into the required format. It's been two weeks since the latest release, and you've just found an issue: someone has found a case where the aggregated data as presented on the CD doesn't match what is shown online.

Since the online version is a relatively new addition to the company's product offering, people are assuming that the CD has always been correct, and the online version has an issue. After some research, you find this is incorrect. In this particular instance, the discrepancy was a side effect of an ID number that got fat fingered during the data entry process. A few subtle flaws in the original aggregation process were also uncovered, but their effect was minimal and generally not material to the end result.

It isn't long before everyone starts panicking at the possibility of a customer noticing the issue. The knee jerk reaction by management is that the online version should be "fixed" – though, of course, you know that the problem isn't with the online version.

From the company's perspective, it would be a costly operational nightmare to try to replace the large volume of CDs that have already been distributed. But then again, it was only a week ago that the department manager gave a long and involved speech on the importance of always striving for quality.

So what do you do? Quality, in and of itself, is not what a company really wants, at least not when the almighty bottom line is at stake. Do you bring this issue up at the next departmental meeting, and point out what you view as a serious problem? Or is it better to lay low and avoid being the bearer of bad news?

For Further Thought

In light of the complex social issues and compounding confusion that can stem from workplace politics, these examples are all relatively simple. In our second installment, we'll consider some circumstances that call for an even greater level of personal judgment, and where a good solution may not exist at all. In the meantime, tell us how you've handled these issues, or others that you have encountered, in the [forums](#).

[Part 2](#)