

# Ethics in Software Testing

By Michael Stahl - March 5, 2018

<https://www.stickyminds.com/article/ethics-software-testing>

## Summary:

When we speak of “test ethics,” the given examples usually are trivial dilemmas. Do we avoid reporting a bug? Do we report that testing is progressing as planned, even though it’s definitely late? These questions are kids stuff: easy because the situation is so black-and-white. But life will present you with complicated cases where the answer is not that obvious.

Many professions have a code of ethics, and software testing is no different. You can read a concise version of the code on the **ISTQB website**—it is part of the foundation-level syllabus.

The version in the syllabus is a terse synopsis of the code and is somewhat hard to read and follow. I recommend reading, if only once, the complete text of **The Software Engineering Code of Ethics and Professional Practice**. In this document, you can easily notice that the code relates not only to technology aspects, but to social and personal aspects as well.

Assuming we are all honest and quality-minded people, why do we really need this code? And if we were not honest, do you think having a documented code of ethics would solve the problem?

It seems trivial and redundant. Just to prove my point, let’s take a short quiz.

1. While testing software, you find a bug that risks the lives of the system’s users. The system is due to be released in two days, and fixing the bug will delay the release. Would you report the bug?

- a. Yes
- b. No

2. You are testing an application that was developed for your company by an external company, under contract. The contractor hints that if you don’t dig too deep and instead report that the software is bug-free, it will have a positive financial impact on your bank account. Do you do as requested?

- a. No
- b. Yes

I assume that in both cases you chose answer (a) without any hesitation. QED! No need for a code of ethics!

When we speak of “test ethics,” what usually comes to our minds are similar trivial dilemmas. Do we avoid reporting a bug? Do we report that testing is progressing as planned, even though it’s definitely late?

These questions are kids stuff: Lie or tell the truth. And the answer is also easy because the situation is so black-and-white. Of course you report bugs; of course you don’t lie about progress.

But life will present you with complicated cases where the answer is not that obvious. Here is an example.

A large company investigated the option of investing in a start-up. The company assigned Joe to coordinate the evaluation work. The more Joe looked into the start-up, he became convinced that the investment is a good idea. Still, to be 100 percent sure, he asked Jane, a software tester, to run a short test cycle on the prototype of the start-up's product. Jane ran some tests and concluded:

- Generally speaking, the solution offered by the start-up works well
- The results are significantly better than other solutions Jane tested in the past
- There was one aspect where the results were strange; more time and more testing are required to conclude how severe this issue is

Jane wrote a report and, before publishing it, asked Joe for comments. To Jane's surprise, Joe asked that she remove the last conclusion. Joe justified his request with an interesting explanation: The goal of the report is to help management decide whether to invest in the start-up. Jane agrees that they should invest, so Joe's argument is, why confuse management with a detail that will cause them to hesitate? Obviously, they do not expect to get the product with zero bugs. They know it's a prototype that calls for more work!

What should Jane do?

In truth, there is no need to construct esoteric cases. There are daily problems that also are not that simple:

- A tester found dozens of bugs in a certain area of the code. The tester knows that the developer owning this code is not to blame for this. The code was inherited from someone else who did a bad job. The tester can report all these bugs, but at a certain point this will risk the developer's career, as the project manager would tag him as the main cause of quality problems. Should the tester report these bugs officially? Or should he pass the reports directly to his colleague, protecting both his peer's job and the product's quality?
- A development manager asks the testers to delay reporting a critical bug by one day. The reason is that today he has a program review and a new critical bug will cause a severe reaction from his managers. It's not that the bug won't be fixed. It definitely will be fixed, but tomorrow.

And so on. It is easy to think up such cases.

I tried to see if consulting the ISTQB Code of Ethics would give a solution to the above conflicts.

For the first case, the Code of Ethics says: "Software testers shall be fair to and supportive of their colleagues, and promote cooperation with software developers." On the other hand, it also has this: "Software testers shall maintain integrity and independence in their professional judgment." From a collegial point of view, especially under the considerations of the case, passing the bugs directly to the developer sounds like the right thing to do. But what about integrity? Independence? The collegial consideration seems irrelevant!

The second case is even more complex. Neither the product, the customer, management, nor the public will suffer any negative impact from a short delay in the report. More likely, it's the other way around: Reporting now will create a harsh, negative reaction from

management, along with the ensuing demand for multiple reporting sessions where everyone would have to prove they are doing their job. Instead, the bug will be dealt with quietly and efficiently tomorrow. Maybe this is actually the correct professional behavior the code hints at—the realization that sometimes it is better not to just follow process without thinking?

It seems that while the Code of Ethics looks good on paper, when we try to apply it to even relatively common daily cases, it does not give clear-cut directives.

Is the Code of Ethics redundant and useless? How can we decide the right thing to do?

I find that the code has merit as long as you do not expect it to have a simple solution for all situations. To start with, the mere existence of the code brings attention to the fact that our work has ethical aspects and that we should expect to face ethical conflicts. The code emphasizes that ethics must be a parameter when making decisions.

It also acknowledges that it is okay to include not just technical considerations, but also “soft” aspects such as supporting peers, fair management practices, and more. These are aspects that, if not for the written code, we would think should not play a role when evaluating a situation.

It is seldom that you will get a clear and final answer directly from the Code of Ethics, but in most cases, it is a good starting point for a discussion.

I consider myself lucky that I personally have encountered very few ethical conflicts in my career as a tester. In all these cases, it happened when someone asked me to do something I thought was inappropriate.

When something like this happens, you are totally shocked. First, because you think the request is so improper. And then, when you realize that on paper the answer is simple (an unequivocal no), but in reality the situation is not that simple—either because of the requestor’s rank, the project’s status, or other circumstances. You realize it’s not so easy to refuse, or you start doubting whether your position is the right one.

It gets even more complicated when the request is from your direct manager. If it’s not something that, once known, will cause the immediate firing of this manager, there is little you can do. Sure, you can go to your manager’s manager. In theory, this is what you should do. And you are probably protected by the company’s nonretaliation policy, but to be realistic, your work environment after such reporting probably will be unbearable.

You can argue with your manager and offer some palatable, more ethical actions, but direct confrontation with your manager and accusation of unethical conduct is a recipe for ruining your work relations. Eventually, you must decide if the case is so problematic that it justifies losing your job over it.

In most cases, I think you don’t really have a choice but to comply and try to minimize the ethical breach. For example, while not reporting a critical bug, add to the report that “the test cycle is still ongoing and it is possible we will still find critical bugs.” At least you manage to convey a clear message that the product is not yet ready for release.

If such requests are happening repeatedly, eventually you won’t have a choice but to expose this to higher management and hope that the situation will be remedied without negative impact to you. In any case, the code hints that a workplace where ethics are being ignored regularly is not a place where you want to be working.

When it's not your direct manager, I found that the best strategy to decide what to do is to consult with others. First of all, talk to your direct manager so that they are aware of the issue and (just as important!) will back you up in case someone claims you acted unethically. Consulting with colleagues from other projects is another effective approach, providing you can trust their discretion.

Someone who is not emotionally involved with the dilemma you face can think more clearly about it. Their unawareness of personal and political aspects will allow them to give impartial advice. Granted, some of it won't be practical because they are unaware of the overall situation, but such conversation will help you understand better the involved parameters and considerations. It will help you see more clearly what's important objectively and what only seems important because of your emotional situation and can be ignored.

Such conversation can help find solutions that are ethical while avoiding conflicts that inflexibility can cause. Additionally, because the advice came from people who are uninvolved and not impacted by whatever you choose to do, it will support accepting a midway approach without feeling you sold out just because doing the right thing was uncomfortable.

If you encounter an ethical dilemma at work, the ISTQB Code of Ethics is a good starting point to figure out how you should act. But consulting trusted colleagues or managers is also a smart idea to be sure you're thinking about every factor. And always let your conscience be your guide.