Part Two

How do software engineers contribute to the good life for others?

There is a second way in which we need to broaden our understanding of engineering ethics. Ethics is not just about avoiding harms, as a narrow focus on preventing catastrophic events might make us believe. Ethics is just as much about doing *good.* 'Doing good' is not something that matters only to missionaries, social workers and philanthropists. To live a 'good life' is to make a positive contribution to the world through your existence, to be able to say at the end of your life that in your short time here, you made the world at least somewhat *better* than it would have been without you in it. This is also how we think about the lives of those who have left us: when we mourn our friends and loved ones, we comfort ourselves by remembering the unique comforts and joys they brought to our lives, and the lives of others; we remember the creative work they left behind, the problems they helped us solve, and the beautiful acts they performed, great and small. *Could a life about which these things could <u>not</u> be said still be a good life?*

If the good life requires making a positive contribution to the world in which others live, then it would be perverse if we accomplished none of that in our professional lives, where we spend many or most of our waking hours, and to which we devote a large proportion of our intellectual and creative energies. Excellent doctors contribute *health* and *vitality* to their patients and medical *knowledge* to their interns and colleagues; excellent professors cultivate *knowledge*, *insight*, *skill and confidence* in their students and contribute the benefits of their *research* to the wider community; excellent lawyers contribute *balance*, *fairness* and *intellectual vigor* to a larger system of justice.

Question 2.1: What sorts of things can excellent *software engineers* contribute to the good life?

(Answer as fully/in as many ways as you are able):

Question 2.2: What kinds of character traits, qualities, behaviors and/or habits do you think mark the kinds of people who tend to contribute most in these ways?

(Answer as fully/in as many ways as you are able):

Part Three

To *whom* are software engineers obligated by their professional ethics? Who is 'the public' that deserves an engineer's professional concern?

The NSPE's **paramountcy clause** asks engineers to recognize that their primary professional duty is to 'hold paramount the safety, health and welfare of the public.' But *who* exactly is this 'public?' Of course, one can respond simply with, 'the public is everyone.' But the public is not an undifferentiated mass; the public is composed of our families, our friends and co-workers, our employers, our neighbors, our church or other local community members, our countrymen and women, and people living in every other part of the world. To say that we have ethical obligations to 'everyone' is to tell us very little about *how* to actually work responsibly as an engineer in the public interest, since each of these groups and individuals that make up the public are in a unique relationship to us and our work, and are potentially impacted by it in very different ways. We also have special obligations to some members of the public (our children, our employer, our friends, our fellow citizens) that exist alongside the broader, more general obligations we have to all of them.

One concept that ethicists often use to clarify our obligations to the public is that of a **stakeholder**. A stakeholder is anyone who is potentially impacted by my actions. Clearly, certain persons have *more* at stake than other stakeholders in any given action I might take; when I consider, for example, how much effort to put into cleaning up a buggy line of code in a program that will be used to control a pacemaker, it is obvious that the patients in whom the pacemakers with this programming will be implanted are the primary stakeholders in my action; their very lives are potentially at risk in my choice. And this stake is so ethically significant that it is hard to see how any other stakeholder's interest could weigh as heavily.

Still, in most ethical contexts, including those that arise in software engineering, there are a variety of stakeholders potentially impacted by my action, and their interests may not always align with each other. For example, my employer's interests in cost-cutting and an on-time product delivery schedule may frequently be in tension with the interest of other stakeholders in having the highest quality and most reliable product. Yet even these stakeholder conflicts are rarely so simple as they might first appear; the consumer also has an interest in an affordable product, and my employer also has an interest in earning a reputation for product excellence, and in maintaining the profile of a responsible corporate citizen.

Of course, while my own trivial, short-sighted and self-defeating interests (say, in gaining extra leisure time by taking reckless coding shortcuts) will never trump a critical moral interest of another stakeholder (say, their interest in not being unjustly killed by my product), it remains true that I myself am a stakeholder, since my actions also impact my own life and well-being. A decision to ignore my well-defined contractual obligations to my employer, or my obligations to my fellow product team members, will have weighty consequences for me. But ignoring the health, safety and welfare of those who rely upon the code I produce has consequences that are potentially even graver – for me as well as for those persons whose well-being I have chosen to discount or ignore.

Ethical decision-making thus requires cultivating the habit of reflecting carefully upon the range of stakeholders who together make up the 'public' to whom I am obligated, and weighing what is at stake for each of us in my choice.

Here is a scenario to help you think about what this reflection process can entail:

Case Study 3

You are a new hire in a product design team for a start-up company that is developing new and more powerful versions of the kind of packet-sniffing and email scanning software systems used by law enforcement agencies and large corporations to monitor data traffic for illegal activities. This kind of software might, for example, be programmed to detect illegal downloads of copyrighted materials, or to flag for review email keywords like 'bomb,' 'steal,' or 'bribe.' You are a young parent of two small children, with parents and friends who are deeply proud of your achievements. You are looking forward to using this first job to cultivate a reputation in your industry for being an excellent software engineer.

One day, you happen to overhear your supervisor chatting with another supervisor about a new contract the company has recently received from a foreign government. You happen to recognize the name of this country as one that is currently run by an oppressive military regime that routinely imprisons its citizens without trial or other due process. In this country, people perceived as political dissidents and their families are often sent to labor camps with deplorable living conditions, without hope of appeal, for an indefinite period. Your own nation has strongly criticized this country's human rights record, and many international organizations as well as the United Nations have condemned its practices.

You realize now that the product your team is working on is part of your company's contract with this government; and in fact, you have been assigned specifically to develop the part of the product that searches for specific keyword strings in private emails, texts, social networking messages, Skype and phone conversations. Reviewing the specs for your task, you realize that your contribution to the product will almost certainly be used to identify for extraction and review conversations between private citizens of this country in which there is *any* specific discussion of their government or its policies, and especially those in which words like 'reform,' 'injustice,' 'corruption,' 'due process' or 'human rights' occur.

Question 3.1: Who are the various stakeholders in this scenario, and what do they each have at stake in your action? ***Reflect carefully and deeply, and answer as fully as possible.***

Type your answer here

Question *3.2*: What do you think is your ethical obligation in this situation? What do you think an *excellent* software engineer would do in this situation? Are they the same thing, or different? Please explain your answer.

Type your answer here

PART FOUR

Why do software engineers have ethical obligations to the public at all? Where do these obligations come from?

As you might expect by now, there is a simple answer to this question that will nevertheless lead us into a far more complex and profound set of considerations. The simple answer is, 'because software engineers are human beings, and *all* human beings have ethical obligations to each other.' Unless you believe, for example, that you have no ethical obligation to stop a small toddler who you happen to see crawling toward the opening to a deep mineshaft[6], then you accept that you have some basic ethical obligations toward other human beings.

What those obligations *are*, precisely, is a matter of **ethical theory**, and many such theories have been developed over the course of human history. Some of these theories developed in folk or religious traditions, others are articulated in scholarly philosophical discourse from the ancient world to today. Among the most well-known and influential types of theory are those of **virtue ethics** found in diverse cultures from Confucian ethics to ancient Greek, Roman and Christian philosophy, along with the **consequentialist** group of theories that include *utilitarianism*, and finally **deontological** theories of ethics that emphasize rules and principles. We will briefly revisit these types of ethical theory in the next section.

Our question here, however, was not '*what are my ethical duties?*' but rather '*why do I have them?*' That is not a question for ethical theory, it is a question of **metaethics**, or the study of where our ethical duties come from and why they obligate us to act as they say we should. Many answers have been given to this question, but before we get lost in a profound philosophical problem, let us remember that in our case we are exploring the special ethical obligations of *software engineers*, which while not wholly independent of our broader ethical obligations as human beings, may have a more clearly identifiable source and justification.

The first explanation of this source involves the concept of a **profession**. What is a professional? You may not have considered that this word is etymologically connected with the English verb 'to profess.' What is it to profess something? It is to stand publicly for something, to express a belief, conviction, value or promise to a general audience that you expect that audience to *hold you accountable* for, and to identify you with. When I profess something, I assert that this is something about which I am serious and sincere; and which I want others to know about me. So when we identify someone as a *professional* X (whether 'X' is a lawyer, physician, soldier or engineer), we are saying that being an 'X' is not just a job, but a *vocation*, a form of work to which the individual is committed and with which they would like their lives to be identified.

[6] This example is adapted from one given by the Confucian philosopher Mencius (in Ivanhoe and Van Norden 2001), who argued that anyone who would be unmoved by the child's peril was not truly human. In the contemporary medical vernacular we would more likely diagnose such a person as a *psychopath* or *sociopath*.

This is part of why professionals are generally expected to undertake advanced education and training in their field; not only because they need the expertise (though that too), but also because this is a important sign of their investment and commitment to the field. When students who have completed an arduous degree program enter the work world, this is taken as evidence that they are sincere in their interest in *this* kind of work, that they understand and uniquely value the contribution that *this* work makes to the world, and that they want their own personal good and sense of self to be enduringly intertwined and identified with the good of their chosen profession. Of course, people do change professions – but not as frequently, or as lightly, as people change mere *jobs.*

So what does being a professional have to do with ethics? How does it create special ethical obligations for the software engineer? First, I stated above that a professional has stated implicitly a desire to have their own good intertwined and identified with the good of their profession. But what *is* the good of their profession? Look back at your answers to Question 2.1 above. Do they suggest an answer to this question?

Consider that members of most professions enjoy an elevated status in their communities; doctors, professors, scientists and lawyers generally get more respect from the public (rightly or wrongly) than retail clerks, manicurists, toll booth operators, and car salespeople. But why? It can't just be the difference in skill; after all, car salespeople have to have very specialized skills in order to thrive in their job. The distinction lies in the perception that professionals secure a ***vital*** **public good**, not something of merely private and conditional value. For example, without doctors, public health would certainly suffer – and a good life is virtually impossible without some measure of health. Without lawyers and judges, the public would have no formal access to justice – and without recourse for injustice done to you or others, how can the good life be secure? Without scientists, the public would be deprived of reliable and carefully tested knowledge – and how can a life filled with ignorance and error be good? So each of these professions is supported and respected by the public precisely because they deliver something *vital* to the good life, and something needed not just by a few, but by us all.

We are nearing the conclusion of our inquiry in this section. We started with the question of why we have ethical obligations as software engineers. Well, software engineering is a profession, and one that like all professions, receives distinctive public support and respect. What do software engineers do to *earn* that respect? How must they act in order to *continue* to earn it? After all, special public respect and support are not given for free or given unconditionally – they are given in recognition of some service or value that actually warrants support and respect. That support and respect is also something that translates into real *power*; the power of public funding and consumer loyalty, the power of influence over how people live and what systems they use to organize their lives; in short, the power to guide the course of other human beings' technological future. And as we are told in the popular *Spiderman* saga, "With great power comes great responsibility." This is a further reason, even above their general ethical obligations as human beings, that software engineers have special ethical obligations to the public they serve.

Question 4.1: Which of the contributions in your answer to 2.1 are related to vital public good(s)? What vital public goods might software engineers help to secure that you did not mention in your initial answer to 2.1?

Type your answer here