Moral Lenses

Abby Everett Jaques
MIT

When you make something, your efforts may go well or badly in many ways. Things may go well in that you may benefit yourself, by furthering your career, or enjoying the feeling of using your skills, or creating something people admire. You may benefit others—your company, your family, your users, or even society at large. Or things may go badly, so that your project is a stain on your career or your reputation, a harm to others, even a danger to the world. Often, things will go well in some ways, and less well in others. Sometimes there are tradeoffs, even painful ones.

Evaluating your project ethically is about understanding all the ways it goes well and badly. It's about tallying up all the things your project does—not just in the narrow sense in which you might conceive it at first, but in as much richness and dimension as you possibly can—and then thinking through the ways in which what it does is good and bad (sometimes both at once!), and for whom.

To assess your project ethically, you'll use four *moral lenses*, each of which offers you a way of looking at a project in order to see what's good or bad about it.

The lenses complement each other: each one gives you a different perspective. Some change the scale at which you're considering the project, as a microscope or telescope does. Some make particular features visible and obscure others, as lenses that filter certain wavelengths do. The lenses provide *goals* and *guardrails*: things to aim for and things to avoid.

Sometimes the lenses will agree; other times something that seems fine when viewed with one lens will seem worrying or even egregious with another. When that happens, you need to think about how the people affected would weigh various harms and benefits—and remember that others may weigh things differently than you do.

A Process for Ethical Engineering

To use the lenses, you first need to think about everything that will be different if your project exists and operates. When you make things, you *change* things; the world is altered. In what ways? Some are small: you may check something off your to-do list; your boss may add something to your list of accomplishments for your performance review; users may gain a new feature in a familiar app. But others are larger: you may get funding for your startup; users may change how they perform an important activity in their lives; old ways of doing things may die off, so that some people lose their jobs, economies both regional and national suffer or thrive; and so on.

Once you've thought of as many ways a project will and may change things as you can, notice all the different people and groups who are affected, and how the effects are similar or different depending on who you're talking about.

Next, use the lenses to get clear about the ways in which particular effects on particular groups are good and bad. The same thing may be good for one group but bad for another, or good in one way and bad in another way for a single group, and so on.

Finally, revisit your project's design: How can you maximize the good and minimize the bad features you've identified? How will you justify your choices to the people and groups affected by them? Maybe your project needs to be quite different. Maybe you shouldn't pursue it at all, because something else would be better.

You can think of this process as involving four steps:

1. **Differences**: Think through all the things your project does, all the ways the world is different with your project in it.
2. **Players**: Catalog the people and groups who are affected by those changes.
3. **Values**: Use the lenses to understand how the differences your project makes for each person or group are *good and bad*.
4. **Design**: Think about which of your design choices affects the good and bad aspects of your project overall. How can your decisions shape the balance and distribution of harms and benefits? Iterate your project in light of steps 1–3 and your new understanding of how your engineering choices are also ethical choices. Then repeat this process with the new

project, until you arrive at a version that seems to achieve the best balance of benefits and harms for all the affected groups.

Step 1: Finding Differences

You make things because you want to change things. You want to provide a better way to do something, or enable something never possible before. You can think of the changes your project makes in *layers of nested systems*:

- **Layer 1**: changes in the system that includes you, your team, your organization
- **Layer 2**: changes in the systems that contain your users, your competitors, the activities your software performs/replaces/changes
- **Layer 3+**: changes in the systems that enclose the first 2 layers: the larger business or institution that contains the activities from layer 2, the industry of which that business or institution is a part; the larger economy of which that industry is a part; and so on.

In each layer, think of as many changes, or potential changes, as you can. Make a list, and when you think of a change in one layer, ask yourself how it would affect the others. The idea is to understand all the changes that will or may happen—big or small; good, bad, or in between—as a result of the existence and operation of your project.

One way to tackle this step is to work forward, starting from what you expect your project to do and imagining how those things will cause further effects. Another way is to work backward: look at similar projects that you or others have created, where you know what (some) of their effects were, and see if or how those effects map to your project, given its similarities and differences compared to the previous one.

You can also consult the work of experts in other fields: each layer involves systems, from companies and other organizations to the health care system, the government, and the economy. Social scientists know a lot about how those systems work, and have important insights to offer. Depending on the specifics of your project, there may be various kinds of experts who've already developed the insights you need. Don't hesitate to seek them out.

A combination of these approaches is often the best bet.

Example: Suppose you make a new app that can perform the tasks of a routine medical checkup. Layer 1 changes may include completing a project you've been assigned by your boss; your organization's being able to go public; etc. Layer 2 changes may include enabling people without easy access to medical care to get checkups outside of doctors' offices or enabling nurses to administer checkups without MDs (one or both of those may have been a design goal from the start, but one may not have been the original plan). Layer 3 changes may include medical practices reducing their number of primary care physicians; higher up we may see insurers requiring the use of the app instead of in-person appointments; medical students shifting away from primary care specialties; changes to the federal budget because of cost savings on Medicare and other federally funded programs, unemployment and/or retraining needs among primary care physicians, etc.

Step 2: Identifying Players

Imagine all the changes you identified in Step 1 were parts of a story, a movie. Who would be in the cast? Notice all the people and groups you've identified—you and your boss, your users and your competitors, and so on. Don't just think about who'd have starring roles; remember that the extras matter too: sometimes very important ethical effects are those that are small at the individual level but matter because they involve so many people, even if those people might initially seem far from the center of the action.

Notice too that sometimes you'll need to subdivide groups: your software may work differently for some subsets of users; it may affect some non-users more than others, etc. Again you can think in layers: at each layer from Step 1, who is affected? Remember to capture secondary effects, too: if some parents are directly affected, there may be important indirect effects on their children, for example. Or if primary care doctors are directly affected, the nurses and other employees in their practices may be indirectly affected.

Add the relevant people and groups, subdivided as needed, to each change on your list from Step 1.

Example: Keep thinking about the medical checkup app. Lots of people and groups came up in describing the changes in each layer: you, people in your company, patients, doctors, medical students, hospital

administrators, insurers, and more. And some of those groups will need to be subdivided: we imagined patients being required to use the app instead of in-person appointments, but that might be true only for less affluent patients; people with very high-end insurance might not be subject to the requirement, for example.

Step 3: Using the Four Lenses

Now that you know how your project makes a difference, and for whom, it's time to get clear on which of those differences are good and bad, and in what ways. (Some differences will be good in one way, and bad in another.) That's what the lenses are for.

I. The Outcomes Lens: When we make something, the state of the world is altered. What changes when your project is created/used/maintained? In what way(s) do things turn out better or worse vs the starting state?

- *Ask*: What good or bad thing, tangible or intangible, does each person or group have more or less of? (health, wealth, power, freedom, security, time, burritos?)
- This lens is about *costs and benefits*.

  Example: Users of the checkup app may save time and money by using it: those are outcome benefits to them. (Though if the app is less good than a doctor at detecting some medical conditions, your users may also lose health by using it. This would be an outcome harm.) Given the problems in the US health care system, an app like yours might be very valuable: you and your company might well make a lot of money. That would be an outcome benefit. On the other hand, some primary care doctors might lose their jobs; that would be an outcome harm.

II. The Process Lens: it's not just what happens, but *how*. Even if the outcome is good, it can still be that something has gone wrong. How is each person or group treated by, and in, this process? Are rules followed? Rights respected? Duties fulfilled?

- *Ask*: Did people have a chance to *consent or refuse*? (Note that EULAs that are too long to read and too full of jargon to understand do not yield meaningful consent.) Were they *deceived* (which undermines their ability to meaningfully consent)? Was someone *used* in ways they might object

to? Was their *privacy* violated? Did people have the kinds of *control* they are entitled to, or were important things out of their hands? Were procedures/rules/etc. followed as people reasonably expect?

- This lens is about the *means* by which outcomes are produced.

  Example: Think about our checkup app again. If many people would prefer to talk to a human doctor rather than using the app, but their insurer won't allow it, then the existence of the app reduces those people's control of how they receive medical care. This is a process harm to those people, even though to other people the additional option may be a process benefit.

  Now imagine the app can monitor people's health without their knowing it (thanks to some fancy hardware/peripherals that check vital signs at a distance using radio waves, plus some data collection about people's activities from their phone accelerometers, calendars, etc, plus more data collection from the Alexas in people's homes that hear what they talk about, their tone of voice, whether they sneeze and cough...).

  Even if people can gain important health information this way—an outcome benefit—it is nonetheless a process harm. Individuals are supposed to get to decide what medical care they get, generally speaking, because control of what happens to your own body is important. Having this app collect data about everyone could also provide important outcome benefits to the population as a whole—say, by supporting medical research, enabling early intervention with outbreaks of infectious diseases, and so on. But again, if people do not have the opportunity to consent or deny consent, there is a process harm. And of course, if you were monetizing all this health data, you or your company might gain that outcome benefit. But that does not negate the process harm.

III. The Structure Lens: how are outcomes *distributed* among people and groups? what are the differences in how people and groups are treated in the process? what are the *patterns* of harm and benefit?

- *Ask*: Is everyone treated *equally*? If not, what is the basis for the inequality? Do the patterns of harm and benefit track historical patterns of advantage and disadvantage, for example by privileging people of a certain race or gender, or do they mitigate historical patterns? Does the distribution of harms and benefits look *fair*, or unfair?

- Remember to consider both outcomes and process when you're thinking about the distribution of harms and benefits: in other words, you want to pay attention not only to the patterns in who ends up better and worse off, but also the patterns in who was treated well or badly as a means to those outcomes.
- This lens is about things like *patterns*, *distribution*, *fairness*, and *bias*.

  Example: Recall the insurers forcing people to use the app instead of seeing a doctor. This may only happen to people who are poor, with inexpensive insurance. The wealthiest people, who have excellent insurance or don't need insurance at all, may have more options. This pattern, where the app gives additional flexibility to people who are already the best off, and limits the control of people who are already disadvantaged, is a structural harm.

  What's more, if the app is better at diagnosing medical problems for some people than others, especially if those differences map to important categories like race or gender, there will be a structural problem even if the app is better than a doctor for many people.

IV. The Character Lens: what kind of 'person' is this project? does it manifest virtue or vice? would a good person create, use, and/or support this project, or not?

- *Ask*: What are the character traits of this project? Does the project (its development, use, operation) manifest virtues like courage, kindness, impartiality, consideration, generosity, and altruism, or vices like cowardice, greed, bias, and selfishness?
- This lens is a bit different from the others. But sometimes it's the most intuitive way to understand ways in which a project can be good or bad.

  Example: What is the character of the checkup app, and those who would create, support, or use it? We can imagine that its developers' goal was to increase access to medical care for those who need it; that would be generous. Even so, if the app is deployed by insurers just to cut costs, without benefiting patients, that's greedy. If the app only works well for certain groups, then it's biased. Sometimes virtuous efforts are exploited by actors operating from vice; sometimes a project can itself seem to

manifest both virtues and vices. Thinking about how your project can support virtuous uses and resist vicious ones can be a good way of working through its ethical dimensions.

Step 4: Make it Ethical by Design

You now know a lot more about what your project is and does, and to and for whom, and the ways in which that's good and bad. Your final step is to think about which features of your project make a difference to the balance and distribution of benefits and harms. What design choices maximize benefits, minimize harms, and do the best job of making sure both are distributed fairly?

At this stage, it's a good idea to think about how you would *justify* your choices to the people and groups affected by them—especially when there are significant tradeoffs. If one of your design choices benefits some people at others' expense, what would you say to those who are bearing the burden? What would they say in reply? One very good way to work through this stage—and the earlier ones, too!—is to talk to as many people from the relevant groups as you can. You don't need to guess; you can *ask*.

Once you've mapped the changes your project produces to particular design choices, and thought through which way to go with those choices by thinking about what you can justify to those affected, you may find you need to rethink your project. Do that, then work through the process again with the new version; keep going till you have a version that doesn't seem to call for more changes. (And of course, the version that turns out to be best could be the one that means abandoning the project: you always need to be able to explain why your project is better than the alternatives, where those alternatives include the status quo.)

> *Example*: One important set of choices for the checkup app has to do with what role actual doctors play in the process. Is the app designed to replace in-person checkups, or is it designed to speed up the checkup process without eliminating the in-person component? If it's the latter, doctors may be in favor of your app; if it's the latter; expect them to object—because the former looks like an outcome benefit to them, since it saves them time and thus allows them either to fit more appointments into their workday or to spend more time communicating with their patients, but the latter is an outcome harm to them, threatening their income and

even jobs. Of course, insurers will see different benefits and harms, and so deciding which way to design the app will involve balancing the needs of each—along with patients and everyone else from Step 2.

Practice Exercises

1. Think of a piece of software you'd like to create. Work through the four steps, using the moral lenses, and think about what the ethical issues are for your project.

- What are the main benefits it will or may provide, and to whom?
- What are the main harms it will or may cause, and to whom?
- How could you maximize the benefits and minimize the harms, and ensure that they are distributed fairly?

2. You've been practicing evaluating the software you create in terms of three important properties: *correctness*, *clarity*, and *changeability*. Correctness is about ensuring that your code does what it's supposed to do. Clarity is about ensuring that those who work with your code understand what they're getting, what they're doing if they use it. And changeability is about ensuring that as circumstances evolve, your code is able to adapt—so that it *continues* to do what it's supposed to do, and what people expect.

All of these properties are important to writing good software, and they often reinforce each other. Working on clarity can help with correctness; ensuring correctness can contribute to changeability; and so on. These properties can also affect the ethical import of a project: that is, failures of correctness, clarity, or changeability can also be failures of outcomes, process, structure, or character. This is because the 3 Cs affect what your project does, whether it does what others expect, and so on.

- How can a failure of correctness become an ethical problem? Use the moral lenses to answer.
- Then do the same for failures of clarity and changeability.

RES.TLL-008 Social and Ethical Responsibilities of Computing (SERC)

Fall 2022