Chapter 4. Engineering for Equity

Written by Demma Rodriguez

Edited by Riona MacNamara

In earlier chapters, we've explored the contrast between programming as the production of code that addresses the problem of the moment, and software engineering as the broader application of code, tools, policies, and processes to a dynamic and ambiguous problem that can span decades or even lifetimes. In this chapter, we'll discuss the unique responsibilities of an engineer when designing products for a broad base of users. Further, we evaluate how an organization, by embracing diversity, can design systems that work for everyone, and avoid perpetuating harm against our users.

As new as the field of software engineering is, we're newer still at understanding the impact it has on underrepresented people and diverse societies. We did not write this chapter because we know all the answers. We do not. In fact, understanding how to engineer products that empower and respect all our users is still something Google is learning to do. We have had many public failures in protecting our most vulnerable users, and so we are writing this chapter because the path forward to more equitable products begins with evaluating our own failures and encouraging growth.

We are also writing this chapter because of the increasing imbalance of power between those who make development decisions that impact the world and those who simply must accept and live with those decisions that sometimes disadvantage already marginalized communities globally. It is important to share and reflect on what we've learned so far with the next generation of software engineers. It is even more important that we help influence the next generation of engineers to be better than we are today.

Just picking up this book means that you likely aspire to be an exceptional engineer. You want to solve problems. You aspire to build products that drive positive outcomes for the broadest base of people, including people who are the most difficult to reach. To do this, you will need to consider how the tools you build will be leveraged to change the trajectory of humanity, hopefully for the better.

Bias Is the Default

When engineers do not focus on users of different nationalities, ethnicities, races, genders, ages, socioeconomic statuses, abilities, and belief systems, even the most talented staff will inadvertently fail their users. Such failures are often unintentional; all people have certain biases, and social scientists have recognized over the past several decades that most people exhibit unconscious bias, enforcing and promulgating existing stereotypes. Unconscious bias is insidious and often more difficult to mitigate than intentional acts of exclusion. Even when we want to do the right thing, we might not recognize our own biases. By the same token, our organizations must also recognize that such bias exists and work to address it in their workforces, product development, and user outreach.

Because of bias, Google has at times failed to represent users equitably within their products, with launches over the past several years that did not focus enough on underrepresented groups. Many users attribute our lack of awareness in these cases to the fact that our engineering population is mostly male, mostly White or Asian, and certainly not representative of all the communities that use our products. The lack of representation of such users in our workforce[1] means that we often do not have the requisite diversity to understand how the use of our products can affect underrepresented or vulnerable users.

CASE STUDY: GOOGLE MISSES THE MARK ON RACIAL INCLUSION

In 2015, software engineer Jacky Alciné pointed out[2] that the image recognition algorithms in Google Photos were classifying his black friends as "gorillas." Google was slow to respond to these mistakes and incomplete in addressing them.

What caused such a monumental failure? Several things:

- Image recognition algorithms depend on being supplied a "proper" (often meaning "complete") dataset. The photo data fed into Google's image recognition algorithm was clearly incomplete. In short, the data did not represent the population.
- Google itself (and the tech industry in general) did not (and does not) have much black representation,[3] and that affects decisions subjective in the design of such algorithms and the collection of such datasets. The unconscious bias of the organization itself likely led to a more representative product being left on the table.
- Google's target market for image recognition did not adequately include such underrepresented groups. Google's tests did not catch these mistakes; as a result, our users did, which both embarrassed Google and harmed our users.

As late as 2018, Google still had not adequately addressed the underlying problem.[4]

In this example, our product was inadequately designed and executed, failing to properly consider all racial groups, and as a result, failed our users and caused Google bad press. Other technology suffers from similar failures: autocomplete can return offensive or racist results. Google's Ad system could be manipulated to show racist or offensive ads. YouTube might not catch hate speech, though it is technically outlawed on that platform.

In all of these cases, the technology itself is not really to blame. Autocomplete, for example, was not designed to target users or to discriminate. But it was also not resilient enough in its design to exclude discriminatory language that is considered hate speech. As a result, the algorithm returned results that caused harm to our users. The harm to Google itself should also be obvious: reduced user trust and engagement with the company. For example, Black, Latinx, and Jewish applicants could lose faith in Google as a platform or even as an inclusive environment itself, therefore undermining Google's goal of improving representation in hiring.

How could this happen? After all, Google hires technologists with impeccable education and/or professional experience—exceptional programmers who write the best code and test their work. "Build for everyone" is a Google brand statement, but the truth is that we still have a long way to go before we can claim that we do. One way to address these problems is to help the software engineering organization itself look like the populations for whom we build products.

Understanding the Need for Diversity

At Google, we believe that being an exceptional engineer requires that you also focus on bringing diverse perspectives into product design and implementation. It also means that Googlers responsible for hiring or interviewing other engineers must contribute to building a more representative workforce. For example, if you interview other engineers for positions at your company, it is important to learn how biased outcomes happen in hiring. There are significant prerequisites for understanding how to anticipate harm and prevent it. To get to the point where we can build for everyone, we first must understand our representative populations. We need to encourage engineers to have a wider scope of educational training.

The first order of business is to disrupt the notion that as a person with a computer science degree and/or work experience, you have all the skills you need to become an exceptional engineer. A computer science degree is often a necessary foundation. However, the degree alone (even when coupled with work experience) will not make you an engineer. It is also important to disrupt the idea that only people with computer science degrees can design and build products. Today, most programmers do have a computer science degree; they are successful at building code, establishing theories of change, and applying methodologies for problem solving. However, as the aforementioned examples demonstrate, *this approach is insufficient for inclusive and equitable engineering*.

Engineers should begin by focusing all work within the framing of the complete ecosystem they seek to influence. At minimum, they need to understand the population demographics of their users. Engineers should focus on people who are different than themselves, especially people who might attempt to use their products to cause harm. The most difficult users to consider are those who are disenfranchised by the processes and the environment in which they access technology. To address this challenge, engineering teams need to be representative of their existing and future users. In the absence of diverse representation on engineering teams, individual engineers need to learn how to build for all users.

Building Multicultural Capacity

One mark of an exceptional engineer is the ability to understand how products can advantage and disadvantage different groups of human beings. Engineers are expected to have technical aptitude, but they should also have the *discernment* to know when to build something and when not to. Discernment includes building the capacity to identify and reject features or products that drive adverse outcomes. This is a lofty and difficult goal, because there is an enormous amount of individualism that goes into being a high-performing engineer. Yet to succeed, we must extend our focus beyond our own communities to the next billion users or to current users who might be disenfranchised or left behind by our products.

Over time, you might build tools that billions of people use daily—tools that influence how people think about the value of human lives, tools that monitor human activity, and tools that capture and persist sensitive data, such as images of their children and loved ones, as well as other types of sensitive data. As an engineer, you might wield more power than you realize: the power to literally change society. It's critical that on your journey to becoming an exceptional engineer, you understand the innate responsibility needed to exercise power without causing harm. The first step is to recognize the default state of your bias caused by many societal and educational factors. After you recognize this, you'll be able to consider the often-forgotten use cases or users who can benefit or be harmed by the products you build.

The industry continues to move forward, building new use cases for artificial intelligence (AI) and machine learning at an ever-increasing speed. To stay competitive, we drive toward scale and efficacy in building a high-talent engineering and technology workforce. Yet we need to pause and consider the fact that today, some people have the ability to design the future of technology and others do not. We need to understand whether the software systems we build will eliminate the potential for entire populations to experience shared prosperity and provide equal access to technology.

Historically, companies faced with a decision between completing a strategic objective that drives market dominance and revenue and one that potentially slows momentum toward that goal have opted for speed and shareholder value. This tendency is exacerbated by the fact that many companies value individual performance and excellence, yet often fail to effectively drive accountability on product equity across all areas. Focusing on underrepresented users is a clear opportunity to promote equity. To continue to be competitive in the technology sector, we need to learn to engineer for global equity.

Today, we worry when companies design technology to scan, capture, and identify people walking down the street. We worry about privacy and how governments might use this information now and in the future. Yet most technologists do not have the requisite perspective of underrepresented groups to understand the impact of racial variance in facial recognition or to understand how applying AI can drive harmful and inaccurate results.

Currently, AI-driven facial-recognition software continues to disadvantage people of color or ethnic minorities. Our research is not comprehensive enough and does not include a wide enough range of different skin tones. We cannot expect the output to be valid if both the training data and those creating the software represent only a small subsection of people. In those cases, we should be willing to delay development in favor of trying to get more complete and accurate data, and a more comprehensive and inclusive product.

Data science itself is challenging for humans to evaluate, however. Even when we do have representation, a training set can still be biased and produce invalid results. A study completed in 2016 found that more than 117 million American adults are in a law enforcement facial recognition database.[5] Due to the disproportionate policing of Black communities and disparate outcomes in arrests, there could be racially biased error rates in utilizing such a database in facial recognition. Although the software is being developed and deployed at ever-increasing rates, the independent testing is not. To correct for this egregious misstep, we need to have the integrity to slow down and ensure that our inputs contain as little bias as possible. Google now offers statistical training within the context of AI to help ensure that datasets are not intrinsically biased.

Therefore, shifting the focus of your industry experience to include more comprehensive, multicultural, race and gender studies education is not only *your* responsibility, but also the *responsibility of your employer*. Technology companies must ensure that their employees are continually receiving professional development and that this development is comprehensive and multidisciplinary. The requirement is not that one individual take it upon themselves to learn about other cultures or other demographics alone. Change requires that each of us, individually or as leaders of teams, invest in continuous professional development that builds not just our software development and leadership skills, but also our capacity to understand the diverse experiences throughout humanity.

Making Diversity Actionable

Systemic equity and fairness are attainable if we are willing to accept that we are all accountable for the systemic discrimination we see in the technology sector. We are accountable for the failures in the system. Deferring or abstracting away personal accountability is ineffective, and depending on your role, it could be irresponsible. It is also irresponsible to fully attribute dynamics at your specific company or within your team to the larger societal issues that contribute to inequity. A favorite line among diversity proponents and detractors alike goes something like this: “We are working hard to fix (insert systemic discrimination topic), but accountability is hard. How do we combat (insert hundreds of years) of historical discrimination?” This line of inquiry is a detour to a more philosophical or academic conversation and away from focused efforts to improve work conditions or outcomes. Part of building multicultural capacity requires a more comprehensive understanding of how systems of inequality in society impact the workplace, especially in the technology sector.

If you are an engineering manager working on hiring more people from underrepresented groups, deferring to the historical impact of discrimination in the world is a useful academic exercise. However, it is critical to move beyond the academic conversation to a focus on quantifiable and actionable steps that you can take to drive equity and fairness. For example, as a hiring software engineer manager, you’re accountable for ensuring that your candidate slates are balanced. Are there women or other underrepresented groups in the pool of candidates' reviews? After you hire someone, what opportunities for growth have you provided, and is the distribution of opportunities equitable? Every technology lead or software engineering manager has the means to augment equity on their teams. It is important that we acknowledge that, although there are significant systemic challenges, we are all part of the system. It is our problem to fix.

Reject Singular Approaches

We cannot perpetuate solutions that present a single philosophy or methodology for fixing inequity in the technology sector. Our problems are complex and multifactorial. Therefore, we must disrupt singular approaches to advancing representation in the workplace, even if they are promoted by people we admire or who have institutional power.

One singular narrative held dear in the technology industry is that lack of representation in the workforce can be addressed solely by fixing the hiring pipelines. Yes, that is a fundamental step, but that is not the immediate issue we need to fix. We need to recognize systemic inequity in progression and retention while simultaneously focusing on more representative hiring and educational disparities across lines of race, gender, and socioeconomic and immigration status, for example.

In the technology industry, many people from underrepresented groups are passed over daily for opportunities and advancement. Attrition among Black+ Google employees outpaces attrition from all other groups and confounds progress on representation goals. If we want to drive change and increase representation, we need to evaluate whether we're creating an ecosystem in which all aspiring engineers and other technology professionals can thrive.

Fully understanding an entire problem space is critical to determining how to fix it. This holds true for everything from a critical data migration to the hiring of a representative workforce. For example, if you are an engineering manager who wants to hire more women, don't just focus on building a pipeline. Focus on other aspects of the hiring, retention, and progression ecosystem and how inclusive it might or might not be to women. Consider whether your recruiters are demonstrating the ability to identify strong candidates who are women as well as men. If you manage a diverse engineering team, focus on psychological safety and invest in increasing multicultural capacity on the team so that new team members feel welcome.

A common methodology today is to build for the majority use case first, leaving improvements and features that address edge cases for later. But this approach is flawed; it gives users who are already advantaged in access to technology a head start, which increases inequity. Relegating the consideration of all user groups to the point when design has been nearly completed is to lower the bar of what it means to be an excellent engineer. Instead, by building in inclusive design from the start and raising development standards for development to make tools delightful and accessible for people who struggle to access technology, we enhance the experience for *all* users.

Designing for the user who is least like you is not just wise, it's a best practice. There are pragmatic and immediate next steps that all technologists, regardless of domain, should consider when developing products that avoid disadvantaging or underrepresenting users. It begins with more comprehensive user-experience research. This research should be done with user groups that are multilingual and multicultural and that span multiple countries, socioeconomic class, abilities, and age ranges. Focus on the most difficult or least represented use case first.

Challenge Established Processes

Challenging yourself to build more equitable systems goes beyond designing more inclusive product specifications. Building equitable systems sometimes means challenging established processes that drive invalid results.

Consider a recent case evaluated for equity implications. At Google, several engineering teams worked to build a global hiring requisition system. The system supports both external hiring and internal mobility. The engineers and product managers involved did a great job of listening to the requests of what they considered to be their core user group: recruiters. The recruiters were focused on minimizing wasted time for hiring managers and applicants, and they presented the development team with use cases focused on scale and efficiency for those people. To drive efficiency, the recruiters asked the engineering team to include a feature that would highlight performance ratings—specifically lower ratings—to the hiring manager and recruiter as soon as an internal transfer expressed interest in a job.

On its face, expediting the evaluation process and helping jobseekers save time is a great goal. So where is the potential equity concern? The following equity questions were raised:

- Are developmental assessments a predictive measure of performance?
- Are the performance assessments being presented to prospective managers free of individual bias?
- Are performance assessment scores standardized across organizations?

If the answer to any of these questions is “no,” presenting performance ratings could still drive inequitable, and therefore invalid, results.

When an exceptional engineer questioned whether past performance was in fact predictive of future performance, the reviewing team decided to conduct a thorough review. In the end, it was determined that candidates who had received a poor performance rating were likely to overcome the poor rating if they found a new team. In fact, they were just as likely to receive a satisfactory or exemplary performance rating as candidates who had never received a poor rating. In short, performance ratings are indicative only of how a person is performing in their given role *at the time they are being evaluated*. Ratings, although an important way to measure performance during a specific period, are not predictive of future performance and should not be used to gauge readiness for a future role or qualify an internal candidate for a different team. (They can, however, be used to evaluate whether an employee is properly or improperly slotted on their current team; therefore, they can provide an opportunity to evaluate how to better support an internal candidate moving forward.)

This analysis definitely took up significant project time, but the positive trade-off was a more equitable internal mobility process.

Values Versus Outcomes

Google has a strong track record of investing in hiring. As the previous example illustrates, we also continually evaluate our processes in order to improve equity and inclusion. More broadly, our core values are based on respect and an unwavering commitment to a diverse and inclusive workforce. Yet, year after year, we have also missed our mark on hiring a representative workforce that reflects our users around the globe. The struggle to improve our equitable outcomes persists despite the policies and programs in place to help support inclusion initiatives and promote excellence in hiring and progression. The failure point is not in the values, intentions, or investments of the company, but rather in the application of those policies at the *implementation* level.

Old habits are hard to break. The users you might be used to designing for today—the ones you are used to getting feedback from—might not be representative of all the users you need to reach. We see this play out frequently across all kinds of products, from wearables that do not work for women's bodies to video-conferencing software that does not work well for people with darker skin tones.

So, what's the way out?

1. **Take a hard look in the mirror.** At Google, we have the brand slogan, "Build For Everyone." How can we build for everyone when we do not have a representative workforce or engagement model that centralizes community feedback first? We can't. The truth is that we have at times very publicly failed to protect our most vulnerable users from racist, antisemitic, and homophobic content.
2. **Don't build for everyone. Build *with* everyone.** We are not building for everyone yet. That work does not happen in a vacuum, and it certainly doesn't happen when the technology is still not representative of the population as a whole. That said, we can't pack up and go home. So how do we build for everyone? We build with our users. We need to engage our users across the spectrum of humanity and be intentional about putting the most vulnerable communities at the center of our design. They should not be an afterthought.
3. **Design for the user who will have the most difficulty using your product.** Building for those with additional challenges will make the product better for everyone. Another way of thinking about this is: don't trade equity for short-term velocity.
4. **Don't assume equity; measure equity throughout your systems.** Recognize that decision makers are also subject to bias and might be undereducated about the causes of inequity. You might not have the expertise to identify or measure the scope of an equity issue. Catering to a single userbase might mean disenfranchising another; these trade-offs can be difficult to spot and impossible to reverse. Partner with individuals or teams that are subject matter experts in diversity, equity, and inclusion.

5. **Change is possible.** The problems we’re facing with technology today, from surveillance to disinformation to online harassment, are genuinely overwhelming. We can’t solve these with the failed approaches of the past or with just the skills we already have. We need to change.

Stay Curious, Push Forward

The path to equity is long and complex. However, we can and should transition from simply building tools and services to growing our understanding of how the products we engineer impact humanity. Challenging our education, influencing our teams and managers, and doing more comprehensive user research are all ways to make progress. Although change is uncomfortable and the path to high performance can be painful, it is possible through collaboration and creativity.

Lastly, as future exceptional engineers, we should focus first on the users most impacted by bias and discrimination. Together, we can work to accelerate progress by focusing on Continuous Improvement and owning our failures. Becoming an engineer is an involved and continual process. The goal is to make changes that push humanity forward without further disenfranchising the disadvantaged. As future exceptional engineers, we have faith that we can prevent future failures in the system.

Conclusion

Developing software, and developing a software organization, is a team effort. As a software organization scales, it must respond and adequately design for its user base, which in the interconnected world of computing today involves everyone, locally and around the world. More effort must be made to make both the development teams that design software and the products that they produce reflect the values of such a diverse and encompassing set of users. And, if an engineering organization wants to scale, it cannot ignore underrepresented groups; not only do such engineers from these groups augment the organization itself, they provide unique and necessary perspectives for the design and implementation of software that is truly useful to the world at large.

TL;DRs

- Bias is the default.
- Diversity is necessary to design properly for a comprehensive user base.
- Inclusivity is critical not just to improving the hiring pipeline for underrepresented groups, but to providing a truly supportive work environment for all people.
- Product velocity must be evaluated against providing a product that is truly useful to all users. It's better to slow down than to release a product that might cause harm to some users.

[1] Google's 2019 Diversity Report.

[2] @jackyalcine. 2015. "Google Photos, Y'all Fucked up. My Friend's Not a Gorilla." Twitter, June 29, 2015. *https://twitter.com/jackyalcine/status/615329515909156865*.

[3] Many reports in 2018–2019 pointed to a lack of diversity across tech. Some notables include the National Center for Women & Information Technology, and Diversity in Tech.

[4] Tom Simonite, "When It Comes to Gorillas, Google Photos Remains Blind," *Wired*, January 11, 2018.

[5] Stephen Gaines and Sara Williams. "The Perpetual Lineup: Unregulated Police Face Recognition in America." *Center on Privacy & Technology at Georgetown Law*, October 18, 2016.