

Issue 005: Ethics

Graham Lee

February 4th, 2019



Welcome to the fifth issue of *De Programmatica Ipsum*, dedicated to the subject of *Ethics*. In this edition:

- Agis Tsaraboulidis¹ provides an overview of the current state of Ethics in technology².
- Adrian takes inspiration from the medical field and proposes the establishment of an “Hippocratic Oath” for software developers.³
- In this issue’s subscriber-only article, Graham explains what needs to be done to raise the standards of ethics in our field.⁴

Enjoy this issue! Please let us know if you have any feedback⁵ and get our free newsletter⁶ to stay updated about new releases. If you want to support us, subscribe⁷ for a month or a year, and let us know if you would like to write with us⁸.

Cover photo by rawpixel⁹ on Unsplash¹⁰.

¹<https://deprogrammaticaipsum.com/user/agis/>

²<https://deprogrammaticaipsum.com/the-current-state-of-ethics-in-tech/>

³<https://deprogrammaticaipsum.com/primum-non-nocere/>

⁴<https://deprogrammaticaipsum.com/what-is-to-be-done/>

⁵<https://deprogrammaticaipsum.com/feedback/>

⁶<https://deprogrammaticaipsum.com/newsletter/>

⁷<https://deprogrammaticaipsum.com/subscribe/>

⁸<https://deprogrammaticaipsum.com/write-with-us/>

⁹https://unsplash.com/photos/9K_tJ7dxYSU?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

¹⁰https://unsplash.com/search/photos/care?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

The Current State Of Ethics In Tech

Agis Tsaraboulidis

February 4th, 2019



Over the course of the last couple of years, more and more press has surfaced about companies and their unethical practices and processes —how they have manipulated or exposed users’ privacy or data for their own gain. This is not new, with many of the reported incidents having happened in 2012-2013 and before.

In 2018 alone, Facebook’s privacy incidents stacked up to 21 and counting. But it would be wrong to target Facebook alone; it is not about a single company but the industry as a whole. Major companies like Uber, Grindr, and Volkswagen were exposed by reporters for their own unethical practices. Though we see article after article published about the act, what I can not find are articles explaining why companies behave that way.

In many of the cases after a company is exposed the founder apologizes (sometimes) and moves on. That way stakeholders, employees, and the public are appeased but actual change is not in the plan of the company. Why? Because then it happens again. It is a never-ending cycle.

As programmers when we deal with a problem, we break it into smaller pieces so we can understand what is happening and make the situation more easily digestible and find a solution. And that is what I would like to do here as well. Here are the main reasons I assume (based on enough news and press releases to count) companies continue to behave that way and do not respect users and their data.

A False Obsession To Change The World

If you ask a founder what the drive behind their company and the product is, 95% of time they will say they want to change the world. By having this very abstract and broad goal as

their north star, they fall into multiple traps without realizing it at the start. By doing so, it removes the ability to think deeply about what it truly takes to achieve that goal. However, the north star is so grand and powerful, it becomes an “at all costs” behavior to reach it.

Leadership: A Focus On Positives And No Plan For The Negatives

One of the most important parts of a company DNA is the ability to adjust the sails after veering off course. It sends a message to the team that we have learned the lesson, we know what caused us to veer off, and we can prevent it from happening again.

In a recent interview, Mark Zuckerberg said:

Facebook was “probably,” he admitted, “too focused on just the positives and not focused enough on some of the negatives.

From Fake-news to Russia’s interference to US elections, these are now big problems that Facebook did not notice in time and now they are paying the price.

By doing so, they enabled all the negatives to pile up without a plan of action. And even if leadership had it, they did not share. But when patchwork on the ship begins because holes are appearing, there will come a point where the ship begins to sink.

Rule of thumb: It is much more efficient and beneficial to assemble a crew to fix the problem rather than patch it up only for another hole to appear. But that is not possible if the crew is not aware of what they are fixing.

Silicon Valley Culture: Profit Over Everything

Silicon Valley is both a magical and mythical place where many of the biggest companies are born and continue to do so. There is no arguing that the drive and spark you get when visiting is like no other. Not to mention the huge valuations, big and unrealistic funding rounds, and culture. But at what cost?

Throwing off the economy? Sexism? Depression? Companies growing at a fast pace (regardless of whether or not they profit) need to continue trucking and investors want (and need) a return on their money. So we turn a blind eye. Theranos, anyone? As long as the wheels keep turning, the press keeps coming, and the users are engaged, the funding continues.

We have founders who focus only on the positives and a culture that promotes profit over anything no matter what. That is a deadly combination and we can understand that through what happened to many companies.

So far we focused on analyzing the problem and the source(s) of it. But what can we do to tackle the actual problem?

1. Legislation Of Tech Companies

There is a huge discussion about legislating tech companies. In many countries, especially in the EU, legislation plays a heavy role in what companies are permitted to do or not. There is less ability to take advantage of users, data exposure (without the proper precautions in place) is not just immoral but illegal, etc. Generally, it is a great idea but in countries like the United States, there is an even more grand problems. The people set to legislate have not the slightest clue how the internet works.

Congressional hearings of Facebook or Google's CEOs are clear examples of this. By simply listening to the questions and responses folks from the committee it is not just clear that they do not understand the basics of the Internet's workings, but lack the desire to learn. Young, educated representatives in office are necessary for a new generation of not only people, but technology, and progress. There is nothing that stunts growth more than mindset.

2. Leadership

With or without legislation, tech organizations and their leaders have a choice and responsibility as to whether they behave ethically or not. So what can leaders do to make a company more ethical?

To kick things off they need to promote good conduct. Lead by example. If the boss never takes a vacation and stays in the office from 8 am to 10 pm, the team will feel inclined to do so. Use their own behavior to create an environment that promotes more ethical processes within the company. By doing that will inspire their own employees to do the same.

In addition to that, leaders need to put unbiased team members with a specific focus on ethics in executive positions; doing so ensures they can intervene when they feel something is unethical or wrong in order to set things straight – creating checks and balances within the company even when leadership is in the wrong.

3. Kick Off Conversations

As an industry, we need to have more conversations about ethics and how to bring change in our workplaces. This cannot be done by a single person but more likely a group of people. Enough employees that have the ability and privilege to take a stand and make their voices heard to company leadership must do so. So the next time you are in a meet-up/conference or hanging out with co-workers, start a conversation. Social media has proven that enough back and forth dialogue can lead to change. That starts with small conversations.

4. Educate The Public

When an article about an incident around privacy comes out, it will most likely be shared around social media (especially within the tech community) and less likely to be shared at a grand scale by folks outside of this bubble. Why? Many people who do not work in tech most likely do not care about such articles. They do not care because they do not know about how important and valuable their data is to companies, and what the repercussions are of said data being exposed.

So we need to educate folks on how to protect themselves and their data on the internet. They need to learn that in many cases they are the product of a company. By educating them the next time a new incident breaks out they are going to be more mindful, understand better what is happening, take action and speak up against the company.

We have a long way to go until we fix this problem but if we do not start doing something, this situation will become worse. As Tim Cook said before, "Privacy is a fundamental human right." Companies have a duty toward humans on the other side of the screen, to build a product that they can love and trust.

Cover photo by Matthew Henry¹ on Unsplash².

¹https://unsplash.com/photos/fPxOowbR6ls?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

²https://unsplash.com/?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

Primum Non Nocere

Adrian Kosmaczewski

February 4th, 2019



Here is a non-revolutionary idea. One that many have had before, and one that many will have after this article. One that might never become true, because of the forces at play and the strength of some contrarian opinions. Here the author proposes the establishment of the equivalent of the “Hippocratic Oath” for the software industry.

Origins

Software as a craft grew out of the needs for computation of capitalism and cold war. Commercial and military computing drove the development of faster and better computers, through a process that ended up putting computers in our wrists, streets, pockets, pacemakers, and in every planet, moon and asteroid in this corner of the galaxy.

Capitalism needed faster computation. Computers, born as a by-product of the Second World War, were perfect for that matter. Computers required software, and this software begat more developers. Some of those developers created faster algorithms, some made simpler programming languages. Some made history, some just made a decent nine to five during forty years.

For a long time, software development did not pay any attention whatsoever about ethics. As practitioners, we cared more about compilers, syntax, libraries, frameworks, IDEs, but we did not ask ourselves whether our software was being used for good causes or not.

Neither Mark¹ nor Jack² did ask themselves that question when they started their famous ventures. They could not care less. Comes to mind that infamous sequence in “The Social

¹https://en.wikipedia.org/wiki/Mark_Zuckerberg

²https://en.wikipedia.org/wiki/Jack_Dorsey

Network:”³

A million dollars is not cool. You know what is cool? A billion dollars.

Hubris

Hubris drives the industry. Not ethics. The current state of affairs is a direct consequence of the lack of questioning in the industry. We have created a broken world.

Our world is made with software. We are the makers of this new world. We must be held accountable, not only for its wonders, but also, and most importantly, for its flaws.

As a profession, we have enjoyed for a long time the privilege of knowledge. We have used the power brought by that knowledge, and we have created one of the youngest professions available to new students these days. We are the new Francmasons, and as such, we must start applying the rules of ethics accordingly.

It is time for the software engineering profession to be driven by ethics. And up to a certain point, regulated.

Accountability

It is time for accountability, for the end of the all-powerful End User License Agreement that removes all liability from the creators of software.

At first sight, however, and given the large applicability of software, it might seem that not all branches of software engineering should be covered by regulation. One could argue that an agency making websites or games might not need to be covered by regulation; companies making software for pacemakers and financial trading, most certainly.

However, the author of these lines does not believe in the auto regulation of markets, and as such, all branches of software should be regulated by state intervention. Such intervention is required in most, if not all parts of modern economies, and software experts are already part of the ranks of governments and judiciary powers.

Definition

Fortunately, the pitiful state of the software industry these days gives a hint to what behaviours could be considered ethical these days:

- Selling consumer data for advertising purposes is unethical.
- Creating consumer products without respect for privacy and security is unethical.
- Releasing software without accessibility features is unethical.
- Racial profiling of any kind is unethical.
- Collaborating with dictatorships is unethical.
- Using language detrimental for any human group is unethical.
- Promoting or simply allowing the spread of fascism or fascist ideas is unethical.

If your software does any of the things above, you are acting unethically through your software, and you and your team are responsible and must be held accountable for it. Each of the developers involved in the creation of such a software (apart from the creators of the libraries and operating systems linked to them,) and in particular those who make a profit out of the sale or promotion of such a software, must be held accountable for breach of ethics.

³<https://www.imdb.com/title/tt1285016/>

Let us be clear: all of the team members must be held responsible. Responsibility cannot and must not be diluted. Hubris can be lethal:⁴

Additionally the overconfidence of the engineers and lack of proper due diligence to resolve reported software bugs are highlighted as an extreme case where the engineers' overconfidence in their initial work and failure to believe the end users' claims caused drastic repercussions.

We need software-savvy lawyers defending online harassment victims. Software-savvy politicians approving new legislation for our software-driven world. Society needs software companies to be held accountable for bugs in critical systems. To be responsible for privacy breaches in consumer systems. To proactively prevent security issues in public infrastructure and national security.

In general, to be responsible for any kind of gross or lesser negligence related to the software put in production.

A First Step

And all of these measures could start with a very simple first step, taken at the end of computer science, programming, or engineering studies: **an ethical oath**, mandatory to all those working in software systems. The very mechanism that Hippocrates found the key to ensure a decent standard of ethics in the medical industry, must be put in action in our own industry, and fast.

Maybe this will be the nudge that will make many managers in software companies, nowadays only attracted to the industry because of good salaries or eye-catching IPOs, to care about quality, accessibility, inclusion, privacy, and security.

Let me be very clear about this: market forces will *not* solve the issues caused by the lack of ethical standards in our industry. Maybe, in a few years time, we are going to witness a new kind of graduation ceremonies, one in which software developers pledge for ethical behaviour.

An Oath For Software Developers

I propose the following one, adapted from the current Hippocratic Oath used in the United States:⁵

I swear to fulfill, to the best of my ability and judgment, this covenant: I will respect the hard-won scientific gains of those software developers and engineers in whose steps I walk, and gladly share such knowledge as is mine with those who are to follow. I will apply, for the benefit of society, all measures that are required, avoiding those twin traps of overengineering and releasing untested code. I will remember that there is art to programming as well as science, and that warmth, sympathy, and understanding may outweigh the knowledge of the developer or the skills of the sysadmin. I will not be ashamed to say "I know not," nor will I fail to call in my colleagues when the skills of another are needed for the proper resolution of a problem. I will respect the privacy of my users, for their lives are not disclosed to me that the world may know. Most especially must I tread with care in matters of life and death. If it is given

⁴<https://en.wikipedia.org/wiki/Therac-25>

⁵https://en.wikipedia.org/wiki/Hippocratic_Oath

me to solve a problem, all thanks. But it may also be within my power to generate another one; this awesome responsibility must be faced with great humbleness and awareness of my own frailty. Above all, I must not play at God. I will remember that I do not merely create a system or implement an algorithm, but I create systems for the highest benefit of society, who will have to use it and who will store their most confidential information within. My responsibility includes these related problems, if I am to solve adequately the problem at hand. I will prevent early code optimization whenever I can, for readable code is preferable to fast code. I will remember that I remain a member of society, with special obligations to all my fellow human beings, those experts in the field as well as those not initiated or knowledgeable in the matters of code and software. If I do not violate this oath, may I enjoy science and art, respected while I live and remembered with affection thereafter. May I always act so as to preserve the finest traditions of my calling and may I long experience the joy of solving the most intricate problems I am faced with.

Conclusion

It is said at the beginning of this article that this is not the first time somebody comes up with such an idea; Graham rightly pointed the author to the ACM Code of Ethics⁶ which serves basically the same purpose, albeit with a longer text.

May any of these ideas find an echo. It is the humble opinion of the author of these lines, that without such mechanism (necessary but not sufficient) we will not be able to start talking about ethics in software engineering properly.

As David Heinemeier Hansson said recently:⁷

Maybe it's time we need an algorithmic oath for programmers: I will program no harm by privacy theft, attention hoarding, radicalization optimization. I will not put engagement metrics above the humans they are extracted from.

Cover photo by rawpixel⁸ on Unsplash⁹.

⁶<https://ethics.acm.org/code-of-ethics/>

⁷<https://twitter.com/dhh/status/1091373596021116930>

⁸https://unsplash.com/photos/IJFnMSGY_bM?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

⁹https://unsplash.com/search/photos/oath?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

What Is To Be Done?

Graham Lee

February 4th, 2019



The title of this article has been used and re-used by socialists throughout history. Luke puts it into the mouths of the crowd listening to John the Baptist. In response to their question, John suggests that those who have surplus clothing and food share with those who do not have enough.

From each according to his ability, to each according to his need.

Russians re-used the title frequently during the decline of the Romanov dynasty. Nikolay Gavrilovich Chernyshevsky was a socialist democrat who wrote a novel promoting industrial collective communes under the title. Then Count Lev Nikolayevich Tolstoy took up the mantle in his response. Tolstoy criticised the idle work ethic of both the poor (who should work harder to improve their lot) and the rich (who are in no position to accuse the poor of being lazy). Perhaps the most famous use of the title is Vladimir Ilyich Ulyanov's, who produced an illegal pamphlet under this heading influenced by Chernyshevsky. Ulyanov's "what is to be done?" set out the case for revolution in Russia. It was a work that set the Russian Empire on a course to become the USSR, and Ulyanov to become its first dictator under his pseudonym, Lenin.

Why Do Anything?

The short question is powerful because of the axiom it implies. To wonder *what* is to be done, we assume that *something* must be done. The situation has become untenable, but we want to know what its replacement should be, and how to bring that about, before we act.

It is also weak, mealy-mouthed, the question demands a programme of action without hinting that the querent intends to do anything about it. Yes, I agree that this absolutist monarchy governed by an ineffectual Tsar should be replaced. You have given me a specific plan of action to bring that about as I requested. OK, good luck with that Vladimir Ilyich, goodbye!

Hmm. Denouncing the parlous state of current affairs. Demanding a plan of improvement. Declining to actually commit to bringing those improvements about? This sounds like a job for software engineers!

An Empire In Decline

The public seems particularly disinterested in advances in computing at the moment, while either masochistically or necessarily becoming more dependent on it. Facebook is in a perpetual state of apology¹. Google have been hit with a record fine². The World Economic Forum expect Bitcoin's value to level out at zero³. A member of the United States congress is discussing racism in artificial intelligence⁴. A cursory news review of software security shows a range of concerns, from security cameras being hacked⁵ to a national early warning network being breached⁶.

Computing "enjoys" a reputation similar to that of genetic modification technology in the 1990s, or nuclear technology in the 1970s. Nobody's really sure what it means, they know it could go drastically wrong. But they also know that it's coming, ready or not, led by the governments and companies who tell us that "progress" is good, whatever that means.

What *Is* To Be Done?

We do need to be part of the discourse and debate around the technologies we use, promote, and inflict upon society. Rather than snarking in our social media bubbles about "the necessary hashtags"⁷, technologists need to be seen, heard, and *understood*. And our positions need to be coherent.

And that means we need to do the thing that is missing in the examples given above: we need to *listen*. Policymakers and journalists are reflecting the concerns that are found in society, and if we want to form coherent and impactful responses to those concerns we need to empathise with the people formulating them. Perhaps we need to walk away from particular activities that society can never condone, rather than convincing them that "making the world more open and connected" is good for them despite their misgivings. We have to listen, learn, empathise and understand *more* than we need to educate, inform and correct. While there is value in the thing on the right, we have to come to value the thing on the left more.

Systems Thinking

Any software artefact is the confluence of three complex systems:

1. the software system itself
2. the team of people who design, build and maintain the software
3. the society of people who find their lives, jobs and opportunities modified by the existence of the software system.

¹<https://www.itv.com/news/2019-01-23/facebook-admits-trust-has-been-damaged-but-promises-change/>

²<https://www.independent.co.uk/life-style/gadgets-and-tech/news/google-gdpr-fine-eu-data-privacy-cnll-amazon-apple-a8740191.html>

³<https://www.cnbc.com/2019/01/23/bitcoin-price-going-to-zero-davos-future-of-blockchain-tech-.html>

⁴<https://www.vox.com/science-and-health/2019/1/23/18194717/alexandria-ocasio-cortez-ai-bias>

⁵<https://www.khon2.com/news/local-news/after-family-s-security-camera-gets-hacked-how-you-can-secure-your-system/1721715522>

⁶https://www.theregister.co.uk/2019/01/08/emergency_warning_network_hacked/

⁷<https://www.theguardian.com/technology/2017/apr/04/amber-rudd-necessary-hashtags-confusion-online-images-videos-home-office>

Much writing on software engineering focuses on the first system (computer science, software architecture, software implementation paradigms) or the second (development methodologies, team organisation, devops). Not so much the third! The NATO 1968 conference on Software Engineering – the event that marked the birth of the field, had a brief discussion on user input into design. Quotes range from:

[J.D.] Babcock [designer of timesharing computing systems]: In our experience the users are very brilliant people, especially if they are your customers and depend on you for their livelihood. We find that every design phase we go through we base strictly on the users’ reactions to the previous system. The users are the people who do our design, once we get started.

to:

[Brian] Randell [IBM, leader of the conference’s working group on design]: Be careful that the design team will not have to spend all its time fending off users.

Empathy

We can argue over the specific time at which empathy in software design entered the software engineering body of knowledge. Barry Boehm’s 1999 paper “Escaping the Software Tar Pit”⁸ introduces the modified golden rule to software engineering:

Do unto others as you would have them do unto you — if you were like them.

But maybe we’re doing Mike Cooley’s 1982 “Architect or Bee?” or Don Koberg’s 1983 “The Universal Traveler” a disservice. Either way, it is only now that many teams are coming to appreciate that the product owner may not have all of the answers about what makes a good product. The fields of User Experience and Design Thinking encourage us to see how people respond to our ideas, to take their criticisms and suggestions on board, and to design the system that *they* will benefit from.

Fundamentally we need to remember that that third system exists. It is society, and we are its designers, its architects, and must bear in mind that we are a minority of its members.

Cover photo by Dmitri Popov⁹ on Unsplash¹⁰.

⁸<http://csse.usc.edu/TECHRPTS/1999/usccse99-533/usccse99-533.pdf>

⁹https://unsplash.com/photos/000djfUJhqQ?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

¹⁰https://unsplash.com/search/photos/socialism?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText