Freedom Voting: Ethnography of a critical software

Camille Akmut

March 3, 2023

**Briefly** Events are reported with a one year distance. Among others, this allows for the possibility of changes having been enabled in the meantime.

This study should be relevant for those wanting to know the conditions in which critical software is created, bound to occupy ever larger segments of life, governing ever expanding and crucial aspects of existence. Explicitly, it is not a cryptographic analysis, does not replace it, nor does the former the ethnographic perspective.

It will also be of interest to those who reflect about the future of free software in the capitalistic society, where it is embedded and plays a complicated and ambiguous role of enabling and subverting it.

**Keywords**: e-voting; electronic voting; free software; capitalism.
1. “Diversity means family”

Bart Rider was a prominent member of the free software community; something he rarely failed to allude to, and often mentioned.

Specifically, he had been a co-founder of one of the Free Software Foundation, he told... (I had never heard of him before, or since.) He spoke of Amandine fondly, and told tales of the ‘old days’. When that, other Free Software Foundation had been created, there was dissent among the Europeans, with Germans going one way and their French counterpart the other. He remarked that sometimes these developments were down to personalities. He halted. Did I know the well-known French free software activist in question? Yes. Was I in contact with him? Yes. (Or more precisely he contacted me.) He thought some, and then arrived at the conclusion that it would be better to talk about all this another time, comfortably around a glass of beer at a pub.

Finally, he showed me a device that served as an illustration for how free software could not -yet- be utilized everywhere, and sometimes exceptions had to be made (it ran on Windows or proprietary software) : a programmable device that let one sign documents electronically. The signatures were valid, he assured. Retrospectively, this either was or should have been the first time I developed some uncertainties about Bart and his company’s mixture of law and technology.

[He later used that device for contract purposes, in an area of the law that does not allow for electronic signatures but requires the written form. Someone must have informed him of this because he then sent 4 letters(!), with the same content. Poor Bart : in his mind he already lived in a 22nd century of wild west electronic signatures and voting... Except not everyone felt the same.]

This was not my first conversation with Bart but the second. In the first one, Bart, my soon-to-be employer put me through a lengthy interview. Pleasantries and niceties were followed by an abrupt transition to an (unannounced) technical interview. This was probably the worst way to go about things, and was unpleasant, but was in line with my overall experience of his company - a mix of amateurism, incompetence and occasional wickedness. He pulled out a piece of code and asked me to explain it. I did. A series of higher order functions and their functions and inputs chained together in the manner typical of the functional style, and an oddity conversion due to the way Python handled types internally (something that would likely not have been needed in a functional language).

In any case, he was satisfied, but one thing that struck as bizarre was that during explanations he kept interrupting and saying he was a "systematic engineer" (an expression that doesn’t translate to most other languages readily). This was the other thing he mentioned often, I noticed, in particular when in a fragile position. "I’m a systematic engineer." (Bart, later used that expression again - "I’m a systematic engineer" - when examining Corona Virus statistics. He constantly minimized the situation and one day expressed the thought that one could also die at any moment in traffic... Someone I knew commented that
this didn’t mean one should walk opposite traffic.)

Hired on the spot. It all smelled of desperation. But, then he backtracked. "Maybe we should do another interview." Evidently he had reconsidered this as well because an e-mail followed shortly: hired, after all.

At this point it is usual to ask if the candidate has any questions. Yet if no question was asked, it was rude and a sign of disinterest. If an actual question was asked, it was "unprofessional". What was the position of the company on diversity? This turned out to be much more complicated than it would have been most anywhere else.

At Google or any of the other devil dens, an experienced, battle-tested, perfectly composed HR lady would have calmly said, with a smile: "We hire irrespective of [blabla] and are proud to support our LGBTQ and black communities through our various initiatives." Banged out like a pro. End of story. No errors. No improvisation. No deviating from the thousand-times repeated company line.

But this wasn’t most anywhere else... For what felt like 20 minutes Bart engaged in monologue-dialogue that bore no evident relationship to the question asked, had no direct link to the subject whatsoever, having I believe mostly to do with the city and families, before arriving at the - odd - conclusion: "For me, diversity means family."

This was really unexpected development, to almost comical degree. It was like answering "For me school means dinner" or "For me ice cream means cookies". It was so absurd, it was hard to contradict or even discuss.

Eventually, he did acknowledge the subject and after some further remarks went: "But, we don’t do quotas for blacks or women, or anything like that."

2. The program : PowerPoint clone is actually e-voting

I had applied for a position related to GPG, but at the last minute was told the job had already (long) been filled, and was instead re-allocated to an application whose title suggested it might be a PowerPoint clone, and on a team - I came to realize - no one internally wanted to be on.

It wasn’t a PowerPoint clone, or rather it wasn’t its main selling point at the height of Coronavirus: one could do slides presentations with it, but that wasn’t its appeal, at least then. Its main selling point during that time was that it let political parties do electronic voting! It was a functionality hidden in the menu under convoluted vocabulary. But, that is what it was: e-voting. Used by many of the major political parties in that country. This came as a shocker. I had read the literature. Had seen the presentations given at "hacker" conferences. The consensus seemed overwhelming: Finger weg! In short, electronic voting - despite being a very alluring idea and convenient - based on all credible sources was not recommended. First day of work: I meet the one that I’ll call here only "His Eminence". Bart headed the company, but His Eminence headed
the project. I try to convey the above in the most diplomatic manner possible: was he aware of the science around electronic voting, and it not being ready for use? He laughed or rather gave a high-pitch noise out of himself; the kind that highbrow people give to suggest they have more knowledge that they do; he might have also mumbled that it was complicated. I always suspected that His Eminence - who was increasingly leaving coding behind to concentrate on doing business and creating relationships with politicians (to whom he peddled and sold his product to) - was aspiring to a sunset career evolution in politics. He had mastered the art of the non-answer, at least, and as such nothing left stood in his way.

On that first day he showed us a lawyer letter, a highlight document of his presentation, that certified his application legally could be used for voting purposes (or so described). It was the letter that he showed to politicians or their aids; who, thus unloaded of the burden of responsibility, could then go ahead and approve it. I should stop to explain how financing works, since this isn’t obvious in the case especially of free software. Briefly, political parties could buy new features, which they would request and then these would be implemented proprietarily - at 1,000-50,000 a pop depending on complexity. Then there was the more classical support and hosting. [I was criticized for helping a, non-paid, user fix their issue as well as for considering a problem encountered by blind users a “high priority” - similarly a community request. I was asked to not work on the latter, and as for the former it was already too late.]

In companies democracy does not exist. “No” does not exist. And, asking too many questions isn’t welcome. This put INNOVATION in a tight spot, as a force in society that purported to create applications to foster democracy.

His Eminence, being the misguided liberal he was, in the presentation of his product was particularly proud of one thing: the far-right party (despite being a legal party, it should be noted) was turned down. God, that filled him so much pride. He was on the right side of history, after all.

On the other hand, FDP (the free marketeers) had been approved, Linke (the left-wing party in crisis, as most are) had been approved, and one of two major centrist parties [I don’t remember which one, i.e. centrist-left or centrist-right.] I asked : what if a communist party came into consideration? Would they, following that divide and implied logic, be turned down also?

He didn’t appear to understand why the question was relevant (e.g. the liberal equivalence routinely drawn between Nazis and Communists as “equally bad” or having “both bad sides”), and possibly did not have any understanding of communism including in his own country (e.g. DKP ban). He answered, the guiding principle would be whether they were "democratic" or not. As if that solved the issue...
Communists followed revolutionary goals I explained - a fact he seemed very surprised by. A decision would be made "case by case" in consultation with Bart.

Then, a long presentation of the "architecture" of the software followed, which would be boring for most readers, and is publicly available and hence won’t be repeated. For our purpose it suffices to say that - like most modern corporate software - it was divided into pieces.

The one extraordinary bit of information was the "by the way" information that lawyer friends of his moonlighted as Golang coders and were responsible for one of these components. I believe already then it was imparted on us that this part, (I think it was the logging section), was not to be touched. (I wondered if these lawyers weren’t the same behind the almighty letter guaranteeing the application was proof secure. Or other lawyers affiliated with their circle.)

Much later, Wolf - a seasoned engineer in his 60s - while going over the same diagram again repeated the same, and that these lawyers/shady/developers were "very picky" about their code. Even while writing this, I find this funny - this stuff is too strange to be made up, and another occasion where reality surpasses fiction. Why be a science fiction writer, when life turns out to be already so weird. The other event of note was the "disappearance" of the only female engineer : I saw her for two weeks, maybe a month, after which she was just gone. No explanations given. Everyone stopped talking about her as if she had never existed, after a few weeks where it was uncertain if she would come back, and there was a sense it was better to not ask questions. Only months later did I learn she had left the company. And, that this was known from the start. For an open source company, INNOVATION did not extend the ‘openness’ to everything. In total 4 people had left the team during, shortly before or after my brief tenure there : Sasha’s friends, the front-end engineer, and her. Considering this was a team of less than 10, turn-over was high. Sasha is the one person I found most sympathetic. He was a graduate student in the middle of finishing his Master’s thesis. Everything revolved around him, certain complex tasks were not done without him (the merging of contributions came up often), despite being the youngest person on the team and the one whose status - as a part-time “working student” - was the lowest, on paper. Being only present on 1 or 2 days a week, his absence was felt. He corresponded to the archetype of the ”10 times” engineer. Sasha was grossly exploited, in my view, like so many others at that company.

It has been my consistent experience that Bart and INNOVATION targeted individuals with weaknesses for employment (from capitalism’s viewpoint) : at one point they even tried to impose an entire week of free labor - conceived as a "trial" - on an unemployed person, who could not refuse due to the conditions set upon unemployment. [A newly hired secretary had uprooted her entire life and her son’s, a teenager in school, in order to move to be with the company. On one occasion, she worked with a broken arm, insisting it was wholly her
own initiative... something strictly illegal.] A few more figures must be introduced now, as they’ll rapidly become relevant. Tony was employed as a software engineer in his older brother’s company i.e. he was Bart’s brother. (And, reminded me of Fredo in the Godfather.) The front-end engineer will be Zuckerberg from now (due to physical resemblance: Mark in his student days — before the barber change and android transformation).

3. Fredo is untouchable and Zuckerberg touchy

Zuckerberg was the one I felt most sorry for: hired out of university, Java educated, OOP indoctrinated, MS Visual Studio equipped, millions of Javascript dependencies and packages ready at the tip of his fingers, ready to be downloaded. He incarnated in all ways the modern “engineer”. (His favorite book was Clean Code by Uncle Bob - really re-inforcing the caricature.) Briefly, front-end engineers (despite some of them being very good) occupy a subordinate position in the hierarchies and are generally regarded as the ”lesser” among the engineers. As opposed to back-end engineers, who are at the top. In the past, I had already observed how touchy front-end engineers could be when it came to that topic. With Zuckerberg it was no different, but it took longer to manifest: He just could not bear discussions that put Javascript or his profession in a bad light. At one point, a news of a major project being breached involving Javascript package management (the infamous npm system) was being widely circulated and thus became a topic of conversation. I had heard of it, another person Fredo/Tony I think too. Oh, did he not like this being brought up! It affected him almost on a personal level.

Whenever these situations arose, instead of letting people talk out an important security issue, he would hush-hush the discussion away with some of his trademark remarks: ”Ok. Ok. We’re done here.” This wasn’t exactly a healthy attitude: why were security discussions inopportune in a project that itself depended on dozens of these dependencies (as anyone could tell by compiling the front-end alone or the whole application.) Hopefully nuclear physics doesn’t have its equivalent of a Javascript issue and front-end people. Everyone became quiet whenever this occurred. Zuckerberg got out of character, then laughed it away, and said ”stop”. There were other exceptions to the rules that happened: Tony in particular, who for a long time I did not understand was Bart’s Brother - only at the very end. But, with that data things started making a lot of sense.

Initially, I had assumed Fredo/Tony was autistic (which made him especially sympathetic). He had frequent outbursts, would only do certain tasks, vehemently refused to do others, left meetings early, and people tolerated all of it, walked as if on egg shells around him. He also used an elaborate setup, as a side note. Behavior that would have gotten anyone else fired many times during the time which I observed it. - If rules were applied uniformly. Which of course they weren’t: the high-earners like Zuckerberg could do anything they wanted short of murder, while Tony was untouchable. Tony was a case of nepotism. The ridiculous stance ”Diversity means family” started making a whole lot of
sense, too, when considered from that perspective and experience.

(His Eminence’s wife was also employed with the company.)

- “Tony, could you do issue #650?”
- “It’s not a 2xx issue, I don’t do those!”

An average meeting would sound like. [Going back to the architecture of the software, Tony only did media files. i.e. the part of the application that handled files, their uploading, processing, etc.]

4. Algebra on text

Zuckerberg carried the front-end engineer stigma heavily and had internalized it: He was obviously interviewing with other companies, and in the interview phase, and thus going through the usual “Pass the interview”-style books; Bolstered by his newfound knowledge, he asked me questions such as did I know any “patterns” and “anti-patterns”? “I know functional programming.” He was stunned. Did not know what to answer. The next day he came back, (had obviously gone through the blogs of the Javascript world), and started with his grilling again. He showed me a random function, wanted to know what it did, as if convinced that he had now found the opportunity to pin me down. None of these activities had any relationship to the project. "It’s an anonymous function written in the new style of ES6." (I did not know Javascript much, but - unfortunately for him - I did enjoy comparing computer languages.) This was far from the end of it. Any reasonable person would have stopped. He continued with his interview practice completely outside of any company structures. Refused to help with the project. "What is a variable?" In the past, I had known a corporate front-end engineer with similar traits. The same unwillingness to share information, the same touchiness, pettiness. I had written a piece of code and wanted to briefly check if this was something he would merge (he had the authorization I lacked): He re-wrote it, modifications including changing the type of one variable from int to text, resulting in code that tried to do algebra operations on text... In other words, something impossible - and which I tried to communicate in a way that would not hurt his ego. He didn’t want to hear any of it. This was something that didn’t even need to be compiled to know it would not work beforehand. Typescript was used, only to ignore the useful information provided by types. I executed it. It, of course, did not work. He concluded, then, that it was an issue of “proper communication guidelines” and repercussions were implied. (Zuckerberg knew Bart’s office better than his own, and being the little rat that he was, often went there to complain I later learned, almost from the beginning.)

Wolf, (with whom I had no significant contact with except for the last days), was in the process of losing his memory. He was in that early stage of onset symptoms where people start to say something, then forget what it was, and unable to continue become angry at themselves, and express their frustration...
out loud - in the middle of conversation. Then they would apologize to the
other person present and ask for a time out.

Significant things were lost, while insignificant ones took on undue propor-
tions: going back to the young man on unemployment INNOVATION had tried
to capture for a week of ‘test’. Free work. He never showed up, citing health
reasons. It was a dirty practice that they used, (he couldn’t theoretically refuse)
and I had no qualms expressing my disgust, that no one should do a week of free
labor. In any case, this was quickly forgotten by all. Except Wolf! Wolf kept
talking about him — a candidate who never showed up — for weeks and months,
and seemed to consider him a dear, esteemed member of the team. I didn’t say
anything. Wolf was the typical 60-year old, burn out, I imagined divorced, life
long corpo. coder that filled me with existential angst. Maybe the memory loss
was not age-related, but stress-induced, as part of some occupational hazard, or
even a side effect illness brought on. He couldn’t concentrate on tasks for long
before veering into gossip about the other employees: Tony was protected as
Bart’s brother, he confirmed. Sasha had had difficulties at university, (needed
to retake a year?), but he insisted was brilliant. Wolf himself readily admitted
he did not fully understand the architecture of the back end - despite working
on it. It was a data store custom made by Sasha and his friends (in lieu of
a ready made, established solution), that he described as a “student project”.
This part of the software (one of the many components alluded to earlier) was
a recurrent object of discussion and critique, mainly to do with performance.
Test cases run on the database made obvious that some entries - which should
have led to failures - instead were let through and could be added. Wolf did
not expect this while doing a demonstration before me, and this was another
matter swept under the rug. Sasha had been offered a job with the company as
“architect” : one of the highest positions a software engineer could have, and
he hadn’t even graduated yet. But, Wolf continued, he hadn’t signed it. I don’t
remember the rest of his confessions, nor did I care. — I did find out who got
the GPG job, however: a systems engineer who had been with the company
for years.

He had made comments about wanting to declare Thunderbird non-free,
based on a definition of “openness” of his own making it seemed. This was
going to end up in the official GPG documentation, of which he was apparently
in charge, until I said that a more reliable indicator was if it used one of the free
licenses and complied. In conversations he talked about mixing personal and
business on his phone. (A Google phone.) Employees were similarly encouraged
to do the same with laptops.

That same sysadmin used long UNIX incantations to copy files from remote
sources and when asked why he did not use the much more common remote
copy program, which did not require a long parameter list, he assured this was
because it did not retain meta-information in the same way (timestamps etc.).
I should not be asking so many questions, he got angry. I still convinced him
to look it up. He laughed, saying it had been so for 10 years. Tee-he-he. In
general, I would describe INNOVATION as a company with a weak free soft-
ware culture. No one that I ever met there, apart from Bart, used the words "free software". Even on matters of free software practice, where it should have excelled, it was behind some of the other companies (which did not advertise themselves as such). It did not use a free forge, notably. When I raised this issue, it was met with unanimous contempt. Bart, obsessed with form rather than substance, after about a month made some public relations reply. If he cared so deeply, what had prevented him from doing something in the many years before. — In the last meeting that I attended I made no secrets about my intentions to leave: "People leave companies for many reasons", I started. Bart was there. He thanked me for my criticism. He left the meeting early, in the middle of things, which was fairly up to highly unusual. Employees were asked what their feelings were. Not one of the dozen present had anything positive to say. Except for Zuckerberg, who said he was shocked and very taken by everything he had heard. Zuckerberg: that sucker, and bootlicker. A newly arrived employee - a young woman recruited straight out of university (for what I assumed was a misery salary) - said she found the meeting "useless" because in the company "no one listened to what others had to say."

The very next day I received a group e-mail written by Bart full of grievances, despite having never received anything close to it. At the end of it, he wished me "a sunny day". A month later, his company was sending me threats of lawsuits. — Bart was vicious, calculated, and calculating. One does not ascend to the top of any community just by being nice. I consider him particularly dangerous. He employed people for poverty wages, and then played the magnanimous philanthropist outside of the company (a way for him to foster his large network; or social capital, as sociologists would say.) My experiences at INNOVATION reflected my feelings about the great state of disarray that I felt the free software community as a whole was in. In an early meeting, a chart was shown to us that clearly established 'Freedom Voting' carried the carriage. It had generated millions of Euros for the company, while expenses were in the hundreds of thousands including salaries. Both waves were color coded such that the difference between income and outcome or expenses was obvious (I calculated it roughly to be 800,000 E). Only one brave soul - among 40 in attendance - dared to ask the obvious question: What happened with the surplus, the margin in between the two which was considerable. His Eminence must have given another one of his non-answers, because I can’t remember it providing any useful additional information or insight. Or he may just have let out another "squeek" sound. INNOVATION was a capitalistic company just like any other. Did the fact that it used free software make it better, or worse? Did it exploit better thanks to free software? On matters of capitalism, the free software philosophy had exactly nothing intelligent to say.
*epilogue*

A critical voting software I came to know was developed in the least likely place, by a colorful group of part-time students, lawyers moonlighting as coders, family hires, burn outs and barely out of university engineers, both full of certainties and insecurities.

— In the middle of all this I made the acquaintance of a politician among the parties mentioned. He was preparing for the next round of big internal elections (about June/July). He said he had never heard of ‘Freedom Voting’. I assured him that he was nonetheless using it. He was in the middle of preparing for a holiday. His left-wing party was going through a crisis having not reached the 5% hurdle, regarded by many as, what it only could be, a major defeat. I found myself now in the position where I was on both sides of the barricades, so to speak: the political battlefront of my politician friend, and the messy software I had somehow gotten myself involved with. He told me about the internal struggles of his party: torn between an older group with anti-capitalist background (communists or former, etc.) and a more prevalent fraction that was representative of the cultural left turn of defeated leftist parties everywhere. I discussed with him the idea that his party should go back to doing old-school economics-based politics if doing plural politics had brought them to their dismal present situation. I’m not sure I got through to him.