

I started this decade as a first-year college student fresh out of high school. I was 17, didn't have a job, didn't have any industry connections, and really didn't know shit. And now you're reading my blog! I would have been proud.

I've told bits and pieces of my story on different podcasts. Now feels like an appropriate time to write down the parts that were most memorable to me.

Every person's story is unique and not directly reproducible. I've benefited immensely from the privilege of being born in an upper middle class family and looking like a typical coder stereotype. People took chances on me. Still, I hope that sharing my story can be helpful to compare our experiences. Even if our circumstances are too different, at least you might find some of it entertaining.

2010

I was born in Russia and I finished the high school there in 2009. In Russia, higher education is free if you do well enough at tests. I tried my chances with a few colleges. I was particularly hoping to get into one college whose students often won programming competitions (which I thought was cool at the time).

However, it turned out my math exam scores weren't good enough. So there were not many options I could choose from that had to do with programming. From the remaining options, I picked a college that gave Macbooks to students. (Remember the white plastic ones with GarageBand? They were the best.)

By the summer of 2010, I had just finished my first year there. It turned out that there wasn't going to be much programming in the curriculum for two more years. But there was a lot of linear algebra, physics, and other subjects I didn't find particularly interesting. Everything was well in the beginning, but I started slacking off and skipping lectures that I had to wake up early for. My gaps in knowledge gradually snowballed, and most of what I remember from my first year in the university is the anxiety associated with feeling like a total failure.

Even for subjects I knew well, things didn't quite go as I planned. Our English classes were very rudimentary, and I got a verbal approval from the teacher to skip most of them. But when I came for the final test, I wasn't allowed to turn it in unless I pay money for hours of "catch up training" with the same teacher. This experience left me resentful and suspicious of higher education. Aside from being a lousy student, I was also in my first serious relationship – and it wasn't going very well either. I was unhappy, but I thought that you can solve this by continuing to be unhappy and "fixing" it. Unfortunately, I didn't have the wisdom to get out of a non-working relationship for a few more years.

Now onto the bright side. Professionally, 2010 was an exciting year for me. I got my first job as a software developer! Here's how that happened.

There was a small venue close to my college that hosted different events. This venue was a "business incubator" – mind you, not a Silicon Valley kind of business incubator – but a tiny Russian one. I have no idea what businesses they "incubated". However, they hosted a talk about software development, and I decided to check it out because I was starving for that kind of content. I didn't know any programmers in real life, and I didn't know meetups existed!

I don't remember what the talk was about now. But I knew the person who gave it was an executive in a Russian-American outsourcing company. I've been programming since 12, so I approached him and asked if they're hiring. He gave me an email, I went through their test exercises, and in a few weeks got the job.

I started at my first job during the summer of 2010. My salary was \$18k/year (yes that's 18 and not 180). This is peanuts in the developed world, but again, this was Russia – so the rent was cheap too. I immediately moved out of my mom's apartment and started renting a room for \$150 a month. I was excited. With my first salary, I bought an iPhone and marvelled at how good the UI was. Summer came and went, and then the second college year started. But it started without me. Now that I was doing actual work and people payed me for it, I lost my last bits of motivation for sitting at lectures and doing exercises. I stopped going there and didn't show up for the midterm exams. I returned my Macbook. The only time I went there again was five years later, to pick up my papers.

A short digression. I'm not saying colleges are worthless, or that you should drop out. It was the right decision for me, but I knew I could fall back on my family (more on that later) when things are tough. I also already had a job.

I had the privilege to be seen as knowledgeable by default due to my background (a guy who started coding early). People who don't fit this stereotype often get a degree just to gain access to that assumed competence. So there's that.

2011

Most of my job was fixing up poor code after cheaper outsourcing companies. Having no industry experience myself, I overengineered every project to try every cool technology I could get my hands on. I even put random Microsoft Research projects in production. Sorry about that. I did some good work too.

My first interesting work project involved a trip. Our client was an investment group in New York. I still don't know anything about investments, but basically they had an email system that received orders, and those orders needed to go through different levels of approval. They had a service that manages that, but the service was extremely flaky, and nobody could figure out how it works. My job was to go onsite, work from New York for a month, and fix the service.

The service was written by a contractor from a cheaper outsourcing company. Nine years later, I still remember his name. The most memorable part of that code was a single thirty thousand line function. To figure out what it does, I had to print it on paper, lay out the sheets on my desk, and annotate them with a pencil. It turned out that it was the same block of code, repeated fifty times with different conditions. I guess someone was paid by the number of lines of code.

I spent that month adding a shitton of logging to figure out what the service does in production, and then rebuilding it from scratch to be less flaky. Working with a non-tech company was a perplexing experience. For example, I couldn't push a bugfix without writing a Word document describing my changes and getting the IT department head to sign off on it. Now that's some code review.

Close to the end of my trip, I went to see a concert at a bar late

at night. The next morning I was supposed to present my month of work to the client. My meeting was scheduled for 9am. Unfortunately, I overslept and only woke up at 1pm that day. My manager apologized for me, and I went home bitterly embarrassed.

There were no repercussions at work. The project was a success overall, and the client knew I'm some weird Russian dude who doesn't know how to groom his hair. But I knew I made a fool of myself. I also wasn't particularly looking forward to more "enterprise projects". Work became a chore.

I was back in Saint Petersburg in Russia. In the summer, the sky there doesn't go dark. During one night of soul-searching, I hopped from a bar to another with a vague sense of unease. Around 7am, a lightbulb went off in my head. I ate a shawarma, took a subway to the office, waited for HR, and quit my job.

My friend was planning a trip to Crimea (before it got annexed) and asked if I would like to join. I packed up a tent and an old Nokia phone that held battery for a week. We camped for two weeks, mostly naked, in a fog of alternative mind states. I barely remember anything from that trip except two episodes.

Once, somebody threatened me with a knife. That person said he would kill me, but he was gone the next day, and everything went on as normal. Another time, I foolishly tried to swim around the cliff alone and almost drowned. I was saved by a rock in the sea that I climbed and passed out on for what felt like an hour.

This trip acted like a hardware reset. My burnout was cured, and I was ready to write code again. (But don't say I told you to almost die to cure a burnout.)

The only problem was... my skills were irrelevant! Yikes.

You see, I was mostly writing desktop software. Has anyone even heard of desktop software? That's not a thing anymore. Either you do backend, or you do mobile, or you do front-end. I didn't know anything about either of them. And I didn't have a job. So I had to move back to live with my Mom. (Thanks, Mom!)

Soon, I saw a post on a social network. It was written by a Russian guy who came back from the Silicon Valley to Russia. He was looking for people who would volunteer to work on his projects for free, in return for him teaching us web development for free. At the time, that sounded like a good deal to me.

I joined this program. I found out quickly enough that there was no real teaching involved: we were given a few tutorials from the web, and we mostly learned from helping each other. Luckily, I could afford to do that for some time while I lived at my Mom's place. I learned Git, basics of Python, Django, a bit of CSS and JavaScript, and some Bash to deploy my changes. Hello Web, here I go.

Nine years later, I'm still unsure how I feel about this program. On the one hand, we were working on his projects for free. On the other hand, we were given full root access, and it was really exciting to be able to push our changes in production and learn from our mistakes. It gave us a structure for learning. It didn't cost us anything, and you could drop out any time. The projects had some social utility due to being around education. This reminded me of open source.

I still feel grateful to this person for setting up this ad hoc "bootcamp" and being my mentor. But I don't want to imply that

working for free is a good way to practice in general. I'm not offering advice here – I am only telling my story.

I built a dashboard where we could track our own learning achievements. My mentor suggested that I pitch it as a product to companies that run courses. My brief foray into playing “startups” was awkward. I didn't know what product I was building, and I pitched different things to different people. Essentially, I ended up making several completely different websites for different clients with a single engine, and earned about \$200 in the process. I wasted months of my time on it, as well as the time of friends who offered to help. I was ashamed of it, and shut it down. The silver lining was that I became a web developer.

But I still didn't have a job.

2012

As a 20 year old web developer, there was only one place I wanted to work at. It was a Russian social media company. Everybody in Russia used their product. That product was very polished. And the team was considered cool. Almost elite.

Their executives often posted about how well-paid their engineers are. The small team of engineers seemed happy with the technical challenges and how the company treated them. In the Russian tech circles, many knew their names.

My mentor introduced me to their CTO, and I got a takehome exercise in JavaScript. It involved building a clone of their typeahead where you could select friends to message. I spent two weeks building it. It was pixel perfect in all browsers. I took care to replicate a similar caching and debouncing behavior.

The onsite interview was a disaster. Clearly, I didn't have the experience at their scale. However, they said they're willing to give me a try if I “understand their product”. They gave me a take-home exercise of designing a logged out state for that social media website. They wanted it to show a picture of a feature phone – many people didn't know the mobile website works on cheap phones.

I spent a week designing that page. I did a lot of tiny details and even hid some Easter eggs in it. I was proud of myself. However, I couldn't find any decent designs of a feature phone that wouldn't look ugly. Instead, I put a beautiful iPhone design there. So aesthetically pleasing, I thought.

Of course, I got rejected. I ignored literally the only hard requirement – why was I so dense? I cried for a few hours because I didn't really want to work anywhere else. I was still living with my Mom and was making no money.

At the time, I was seriously doubting my skills. Many things seemed “magic” to me. I started having doubts about whether dropping out was a good idea, after all. I signed up for an iOS development course on iTunes U. I also signed up for two courses on Coursera: Compilers and Machine Learning. Maybe they would make me a “real programmer”.

It was lonely to go through these courses on my own. I organized a tiny meetup with a few people from our web development “bootcamp”. We would gather and watch different online courses together at my mentor's coworking space.

A month into it, I got an email. Someone was looking to hire a developer, and he heard about me from a person who went to my

meetup. I was sick with mono and ignored the email, but this guy kept emailing me. He wanted to skype.

Roman Mazurenko was not a typical startup founder. Roman was interested in DIY publishing. Together with a couple of friends, for a few years, he somehow made Moscow cool. He organized parties and posed for fashion magazines. I didn't know what to expect. But Roman was very down-to-earth and fun to talk to. His dream was to build a DIY publishing platform like in this concept video. I would have to move to Moscow and learn iOS development on the go. (By the way, the video guy is not Roman but a friend, and the app in the video is a fake animation made with Flash. Roman was great at crafting smoke and mirrors.)

I said yes.

I didn't finish my Compilers and Machine Learning courses. I learned enough to know these topics aren't magic. After that, I lost most of my interest in them.

2013

By 2013, my salary was \$30k/year – almost twice what I made at my previous job. While low by US standards, it was pretty decent in Russia. I also negotiated some stock at Stampsy (spoiler alert: it ended up completely worthless).

Our team had five developers and two designers. We started by developing an iPad app, but neither of us had any real knowledge of iOS. I remember my relief when a teammate figured out how to implement the animation we needed for the first time. Until then, I thought we were doomed.

For a few months, I literally lived in our office. Looking back at this period, I'm not proud of my life-work balance, and it was not healthy. However, I learned more during these months than in two years before, and I don't regret them.

Eventually, I moved out of the office. I've started to live in the same flat as Roman. My room cost me \$1k/month. It was a spacious flat in the only area of Moscow that I enjoyed, and it was only five minutes of walk from the office.

We thought that some of the code we wrote might be useful to others. So we started publishing those pieces on GitHub. We didn't expect anything grand, and even getting a couple of contributions was really nice. The most popular project I worked on during that period has 30 stars. To us, that was a lot.

A designer on our team introduced me to Bret Victor's talks – particularly, to *Inventing on Principle*. I thought it was a very good talk. A very good one.

In April, we released the iPad app we've been working on. Apple reached out to our team and asked for assets to feature it in the App Store. We were over the moon. It stayed featured for weeks, and people started using it.

Our excitement quickly wore down as we realized there was no product market fit. The app was designed to create beautiful magazine-like layouts, but nobody had any beautiful content on an iPad. Also, the iPad camera had a horrible quality. The product didn't make any sense. How could we not realize this?

My personal relationship was falling apart too. We weren't a good fit, and mostly clung to each other out of fear of being alone. We finally broke up.

For a few months, I didn't talk to people from our shared mutual circle and focused on work. But I realized that I missed one particular friend. I wrote to her, and she said she missed me too. I arranged a plan for a trip together.

I caught a cold. As the day of our would-be trip got closer, I felt worse, but I was hoping that maybe I'd be okay. When my train from Moscow to St. Petersburg had arrived, I clearly had a temperature. She said to come to her place anyway. She made me some hot tea, gave me warm socks, and we kissed. I moved in.

2014

For me, 2014 was the year of React.

After a brief existential crisis, we abandoned the iPad app and decided to pivot to a webapp. That meant I had to learn JavaScript, this time for reals. We built a decent prototype with Backbone, but interactive parts were a pain to code.

My colleague saw React but initially dismissed it. But a few months later, he told me React wasn't actually that bad. I decided to give React a try. Ironically, the first component I converted from Backbone to React was a Like button.

And it worked well. Really well. It felt unlike anything I've seen. React wasn't a hard sell for the team. Over the next year we gradually converted all of our UI to React while shipping new features. React and the principle of unidirectional data flow allowed us to develop faster and with fewer bugs.

We started a private beta, and some photographers enjoyed creating visual stories with it. It was something between Medium, Pinterest, and Tumblr. There wasn't a lot of traction, but it wasn't a complete failure like the iPad app.

The only problem with using React was that there was virtually no ecosystem. When we just started, there was only one router (which was not React Router), and we couldn't figure out how to use it. So we made our own. After React Router came out, we adopted it and added a feature we needed for our product.

There was no drag and drop solution for our use cases, so I ported my colleague's library to React. I made a helper to manage document titles. Wrote another library to normalize API responses. Jing Chen from the React IRC channel suggested the core idea, and it worked! Little did I know that in a few years, Twitter would build their new website with this library and maintain it.

I wanted to contribute to React itself, too. I reached out to Paul O'Shannessy and asked if there were any pull requests I could work on. I finished my first task in a few days, but it didn't get merged until a few months later. Big project release cycles, and all that. I was frustrated by the slowness in response so I put effort into the ecosystem instead. In retrospect, that was a lot more impactful. In 2014, I did some of my first public speaking. I gave sort of a lecture about React at our office. It ended up going for two hours, and I'm still surprised that most of the people who showed up were polite enough to stay to the end.

Later, I signed up to give a talk at the BerlinJS meetup. My topic was "React and Flux". I didn't practice my talk, and I only went through the first half when my time ran out. People rolled their eyes, and I finally learned my lesson. From that point on, I would rehearse every talk from three up to fifteen times.

In 2014, I got my first email from a Facebook recruiter. I missed it in my inbox and only found it many months later. We still chatted eventually, but it turned out that hiring me in USA wouldn't be easy because I didn't have enough years of experience and I dropped out of college. Oops.

There was one project I started in 2014 that was particularly dear to me. Like most important things in my life, it happened randomly. I was converting our app from require.js to webpack to enable code splitting. I read about a bizarre webpack feature called "hot module replacement" that allowed you to edit CSS without reloading the page. But in webpack, it could work for JavaScript too.

I was really confused by this feature so I asked about it on StackOverflow. Webpack was still very new, and its creator noticed my question and left a response. It gave me enough information to see I could tie this feature with React, in the spirit of the first demo from Bret Victor's talk I mentioned earlier.

I wrote an extremely hacky proof of concept by editing React source code and adding a bunch of global variables. I decided I won't go to sleep until it works, and by 7am I had a demo I could tweet. Nobody cared about my Twitter before that, but this received some likes and retweets, and those 20 retweets were hugely validating. I knew then I'm not the only one who thinks this is exciting. This proof of concept was a throwaway effort and I didn't have time to keep working on it at my job. I took a vacation, and finished off the prototype there.

Quick disclaimer: again, I'm not saying you "need to" work nights or vacations. I'm not glorifying hustle, and there are plenty of people with great careers who did none of that. In fact, if I were better at time management, I could probably find a way to squeeze those non-interrupted hours in my regular day, or to learn to make progress with interruptions. I am sharing this because I'm telling my story, and it would be a lie to pretend I did everything in a 40 hour week.

2015

Our product went out of a private beta. It was growing, but slowly and linearly. The company was running out of funding and struggled to raise more money. And I wanted to spend more and more time on my open source projects.

I also wanted to give my first conference talk. Naturally, I wanted to talk about hot reloading, but I knew somebody already mentioned it at ReactConf, and I thought people wouldn't be excited about it. I decided to add some spice to my talk proposal by adding "with time travel" – again, inspired by Bret's demo. The proposal got accepted, and for a few months I didn't think much about it.

In April, my salary got delayed by several weeks. It went through eventually, but I realized it's time to look for a new job. I found some company using one of my projects, and they agreed to sponsor my work on it for a few months.

My girlfriend asked if I wanted to get married. I said I thought I'd get married late in my 30s. She asked: "Why?" I couldn't really find any justification for waiting so we soon bought rings and got married. Our wedding has cost us \$100.

The deadline for my talk was approaching. But I had no idea how to implement "time travel". I knew that Elm had something similar, but

I was scared to look at it because I worried I'd find out time travel can't be implemented well in JS.

At the time, there were many Flux libraries. I've tried a few of those, notably Flummox by Andrew Clark, and I had a vague sense that making hot reloading work with Flux would also let me implement time travel. Sunil's gist led me to an idea: a variant of Flux pattern with a reducer function instead of a store. I had a neat name in mind for it already. And I really needed it for my talk!

I implemented Redux just in time for my demo of time travel. My first talk rehearsal was on Skype. I was sweating, mumbling, and running over it too fast. At the end of my rehearsal, I asked the organizer if my talk was any good. He said "well... people like you" which I took as an euphemism for horrible.

I asked a designer friend from the startup I just quit to help make my slides pretty. I added animations and transitions. The more polished my talk looked, the calmer and more confident I felt. I practiced it a dozen times.

I flew to Paris for my first technical conference. This was probably the happiest day of my life. For the first time, I put faces next to avatars. There were UI nerds and my personal heroes around me. It felt like going to Hogwarts.

My talk almost didn't happen. In the morning, I found that my laptop refused to connect to the projector. I only had a few hours left.

Christopher Chedeau was kind enough to lend me his laptop, and I transferred my live demo setup to his computer (except for the Sublime license, as you may know if you watched it).

At first, my demo didn't run on Christopher's laptop because we had different Node versions. The conference WiFi was so bad that downloading another Node version was a non-starter. Luckily, I found an npm command that rebuilds the binaries. It saved my demo. I gave my talk with his computer, and it went well.

I met many people in the audience who I already knew from Twitter. One of them was Jing Chen. I remembered her from our IRC chat on the React channel, and came to say hi. She asked whether FB recruiters contacted me before, and I said I couldn't get a US visa. Jing asked whether I'm interested in working at the London office, and I had no idea there even was a London office! I called my wife and asked if she's up for moving to London. I thought she'd hate the idea, but she immediately said yes. So I said yes to an interview.

There were four people from FB at the conference, so Jing arranged a full interview right at the conference hotel. It was a regular interview process, except it was in Paris and everyone was sweaty because it was so hot outside.

Everything happened so spontaneously that I neither had time to prepare, nor to get nervous. At one point I freaked out and panicked because I couldn't write three lines of code that swap two items in an array. I asked Jing to look away for a few seconds. She said "I know you can swap two items", and that gave me the confidence to finish the answer and make it through the interview. I probably didn't pass with flying colors, but I got the offer.

My talk got really popular. Andrew Clark has already deprecated Flummox – the most popular Flux library – in favor of Redux, which he co-wrote with me. People started putting Redux in production. New learners were confused by the README, which was written for early

adopters who had all the existing context. I didn't have a job, and it would take me several more months to get a UK visa.

I started a Patreon to sustain my projects for a few months – and specifically to write Redux documentation, create small example apps, and record some free videos about it. I raised about \$5k/month on Patreon which was more than any salary I made by that point in my life. Folks from Egghead sent me some mic gear, and I recorded my “Getting Started with Redux” course. I can't watch it without cringing today, but it has been very popular and made me good money (around \$3k/month in royalties) for quite a while – even though it was free.

FB took care of handling most of the immigration process. My wife and I only had to fill some papers, and I had to take an English exam and do some health checks. FB did most of the work to relocate us, including moving our cat from Russia to UK (which cost around \$5k). I was hired at engineering level 4, with initial base salary of \$100k/year and an initial restricted stock unit grant of \$125k vesting over four years. I also had a signing bonus of \$18k which helped a lot as we were settling in. (By the way, tech salaries are lower in UK than in US.)

We arrived to London at the end of November 2015. We've never been to London before. We took a black cab from the airport. We couldn't figure out how to turn off the heating in the cab for ten minutes so we got very sweaty and couldn't see anything through the window. When we turned off the fan and the window cleared up, our eyes were wide like saucers. London was beautiful.

The next day, Roman Mazurenko got hit to death by a careless driver. He just got his US visa approved, and he came to Moscow to pick up his documents. He once told me that there's something devilish about Moscow. It doesn't just let you go. I would not see my friend again. Not in 2015, not ever.

Roman has had a sort of a digital afterlife, courtesy of his friends. I know for a fact he would've loved the irony of having a two point five star App Store rating.

2016

New job. New city. New country. Different language. Unfamiliar accent. Big company. Orientation. Meeting rooms. Projects. Teams. Documents. Forms. Shit. Fuck. Fuck. Fuckety Fuck, Shit, Oh Dear, Blimey!

I barely remember the first few months. I was in a constant state of stress from trying to understand what people are saying without subtitles. What the hell is my manager telling me? Is it rude to ask to repeat again? Spell it for me?

What, I need to call a lady in Scotland to get a national insurance number by phone? I don't get a word she's saying. What even is the national insurance? Why do I have a “zero” tax code and why is my salary less than I expected? Wait, people actually pay taxes here? What do I do when I'm sick? What's NHS?

During my first trip to US in 2016 (part of onboarding), I forgot to eat the whole day, drank a lot of coffee, and had a full-fledged panic attack trying to explain how hot reloading works to a colleague. We called an ambulance, and I got a \$800 bill (thankfully, paid by FB – or at least I don't recall paying it myself).

Relocation was nerve-wracking even though the company handled most of the difficulties. I thought I did everything in the onboarding instructions, but I forgot to register with the police. (I confused this with registering at the post office, which we also had to do.) I only found out that we screwed up months later, and we were told it might affect our visas. Luckily, it didn't so far.

Originally, I was supposed to join the React Native team in London. Usually, we hire people to go through Bootcamp and pick a team, but I didn't have that freedom. I was preallocated. However, I wasn't very excited about React Native in particular. I talked to Tom Occhino (he managed the React team at that time), and he suggested that I can join the React Core team (based in US) as the only UK member. I was already used to remote work from open source, so I agreed.

In 2016, there was a React boom, but everybody made their own "boilerplate" with a bundler, watcher, compiler, and so on. React became synonymous with modular JavaScript, ES6, and all the tooling complexity. Christopher Chedeau suggested to prototype a command-line interface for getting started with React. We made the first version in a few weeks and released Create React App.

2017

Egghead courses continued to bring steady side income with royalties. I didn't think twice before spending them on food delivery or nice clothes.

It's only in 2017 that I came to realize that these royalties are taxable as foreign income, and that I owe Her Majesty something like \$30k. Oops. Like an adult, I got an accountant. Fixing this mess depleted all my savings again.

At work, we spent most of 2017 rewriting React from scratch. You know the result of it as React 16. Sophie tells the story of how we did it here.

Aside from taxes, there wasn't a lot happening in my personal life. I was still settling in. I got less shy dealing with bureaucracies. I could make and take phone calls without freaking out. I watched movies without subtitles. I learned how to deal with NHS and with private insurance. I stopped doing side projects.

2018–2019

The last two years have been a blur. I'm still too close to them to have a clear perspective on what was important.

Professionally, our projects are as demanding as ever. If you follow React, you know about some things we have been working on. I've grown as an engineer, and still have much to learn. Our London team has grown – I'm not alone now.

People occasionally recognize me. This is humbling. Someone once recognized me in a sauna and started complaining about React. Please don't be that person.

I got promoted. I started this blog as a side project. I have another side project coming, which I've been musing about for the better part of these two years. I met more people from the internet and put more faces to avatars. This is fun.

I always knew that I liked building UIs. I got hooked on Visual Basic. I spent this decade building UIs, and then building a way to build UIs. And then talking about it and explaining it. But I realize now that my drive to explain things is just as important to

me as my drive to build. Perhaps, even more important.
I look forward to doing more of that in the next decade.
Or, should I say, this decade?
Welcome to the twenties.