

Contents

October 2021	7
October 18, 2021	7
October 20, 2021	7
October 21, 2021	7
October 22, 2021	8
October 26, 2021	8
October 27, 2021	8
October 28, 2021	9
October 29, 2021	9
November 2021	11
November 01, 2021	11
November 02, 2021	11
November 03, 2021	11
November 04, 2021	12
November 05, 2021	13
November 08, 2021	14
November 09, 2021	14
November 10, 2021	14
November 11, 2021	15
November 12, 2021	15
November 15, 2021	16
November 16, 2021	16
November 17, 2021	17
November 18, 2021	17
November 19, 2021	18
November 22, 2021	19
November 23, 2021	20
November 24, 2021	20
November 25, 2021	21
November 26, 2021	21
November 27, 2021	21
November 28, 2021	22
November 29, 2021	22
November 30, 2021	22
December 2021	24
December 01, 2021	24

December 02, 2021	24
December 03, 2021	24
December 04, 2021	25
December 05, 2021	26
December 06, 2021	26
December 08, 2021	27
December 09, 2021	28
December 10, 2021	29
December 12, 2021	30
December 13, 2021	31
December 14, 2021	32
December 15, 2021	33
December 16, 2021	34
December 17, 2021	34
December 20, 2021	35
December 21, 2021	36
December 22, 2021	38
December 23, 2021	40
December 27, 2021	41
December 28, 2021	42
December 29, 2021	42
December 30, 2021	43
January 2022	44
January 03, 2022	44
January 04, 2022	44
January 05, 2022	45
January 06, 2022	46
January 07, 2022	46
January 10, 2022	47
January 11, 2022	48
January 12, 2022	48
January 13, 2022	48
January 14, 2022	49
January 15, 2022	50
January 16, 2022	50
January 19, 2022	51
January 20, 2022	51
January 21, 2022	51

January 24, 2022	52
January 25, 2022	52
January 26, 2022	53
January 27, 2022	54
January 28, 2022	55
January 31, 2022	56
February 2022	57
February 01, 2022	57
February 02, 2022	57
February 03, 2022	59
February 04, 2022	60
February 07, 2022	62
February 08, 2022	63
February 09, 2022	64
February 10, 2022	65
February 11, 2022	66
February 12, 2022	68
February 14, 2022	68
February 15, 2022	69
February 16, 2022	70
February 17, 2022	72
February 18, 2022	74
February 21, 2022	76
February 22, 2022	77
February 23, 2022	78
February 24, 2022	79
February 25, 2022	80
February 28, 2022	81
March 2022	83
March 01, 2022	83
March 02, 2022	83
March 03, 2022	84
March 04, 2022	84
March 05, 2022	85
March 07, 2022	86
March 08, 2022	87
March 09, 2022	87
March 10, 2022	88

March 11, 2022	89
March 15, 2022	89
March 16, 2022	90
March 17, 2022	91
March 18, 2022	93
March 21, 2022	94
March 22, 2022	95
March 23, 2022	95
March 24, 2022	96
March 25, 2022	96
March 28, 2022	96
March 29, 2022	97
March 30, 2022	97
March 31, 2022	98
April 2022	100
April 01, 2022	100
April 04, 2022	100
April 05, 2022	101
April 06, 2022	102
April 08, 2022	103
April 11, 2022	103
April 12, 2022	104
April 13, 2022	104
April 15, 2022	105
April 21, 2022	105
April 22, 2022	106
April 25, 2022	107
April 26, 2022	108
April 27, 2022	108
April 28, 2022	109
April 29, 2022	109
May 2022	111
May 02, 2022	111
May 03, 2022	111
May 04, 2022	112
May 05, 2022	113
May 06, 2022	114
May 10, 2022	115

May 11, 2022	115
May 12, 2022	116
May 13, 2022	117
May 16, 2022	118
May 17, 2022	119
May 18, 2022	120
May 19, 2022	120
May 20, 2022	121
May 23, 2022	121
May 24, 2022	121
May 25, 2022	122
May 26, 2022	122
May 27, 2022	122
May 31, 2022	123
June 2022	124
June 01, 2022	124
June 06, 2022	124
June 07, 2022	125
June 08, 2022	125
June 09, 2022	126
June 10, 2022	126
June 13, 2022	127
June 14, 2022	128
June 15, 2022	128
June 15, 2022	129
June 16, 2022	129
June 21, 2022	130
June 22, 2022	130
June 23, 2022	131
June 24, 2022	131
June 28, 2022	132
June 29, 2022	133
June 30, 2022	134
July 2022	136
July 01, 2022	136
July 06, 2022	136
July 07, 2022	136
July 08, 2022	137

July 11, 2022	138
July 12, 2022	139
July 13, 2022	140
July 14, 2022	141
July 15, 2022	142
July 18, 2022	142
July 19, 2022	143
July 20, 2022	144
July 21, 2022	145
July 22, 2022	145

October 2021

October 18, 2021

I'm in a new company, "You Robot". Started today. So far, so good.

Had a HIPAA training (I kind of knew it, but there are details, and, as usual, lawyers have distorted everything: where's Portability, for instance?)

Okay, whatever. No time left for anything else. Installed homebrew, that's it.

They gave me a link to docs. And they assigned to me a task, it's called PI. We will see.

Sent them a pic of my password. Like, here's my mug shot, and here's me myself.

October 20, 2021

Nothing to say today. I was configuring the environment.

We need three VPNs here. Well, one is for Production, I don't need it right now. Another is for Staging, and yet another for Dev.

I've already installed tons of things. Had to install sdkman. By the way, the wiki where I'm getting information from, they allow me to edit it, so I'm editing it ad libitum.

I've missed one meeting, but Arie told me that since I have nothing to say for now, there's no need to attend. Wow, it's pretty informal.

I have also promised one colleague to help with Scala. Wrote a short learning plan for him.

October 21, 2021

No news so far. I'm configuring the environment. They told me to us the Ultimate version of IntelliJ, and then "expensify" it.

Okay, not bad. So far I'm kicking all these things, configuring them.

At 1pm (our time) there was a meeting, a scrum one. Everybody's so nice! I have to be like this too, not like Gershman in prison.

Пока никаких новостей. Конфигурирую; интеллиджей велели взять ultimate, за свои деньги, а потом expensify it.

No, the work day still continues, but it's 8pm here already, and I'm having a dinner.

October 22, 2021

We had four meetings today. I was just sitting there quietly. When they asked me to introduce myself, I introduced myself.

Otherwise, I can now see the project in IntelliJ. The code is JUST HORRIBLE. But the people look peaceful in this hospice.

A “new hires” page suggest to get through a “Google Machine Learning Training”. So I do. It’s at the kindergarten level. But I’m listening, why not. Gradient Descent for Dummies.

In the evening I took another look at the code. Arie asked me to have timeouts in the services. Sure, can do. But the code is horrible. And bad spellings all around. Well, what can I do. Horrible code. Ok, will have to deal with it.

October 26, 2021

Not much to say today. Spent my day trying to get access to Jenkins and to Artifactory. Now I have four VPNs installed, and they still want me to click a button on my phone. But still no access.

Why do I need Jenkins? The problem is, there’s a wrong version of Scala plugin for Maven. To be more precise, the old version of the plugin doesn’t work (the compiler *crashes*). And the new one, according to Sebastian, does not work on Jenkins. And they send me here and there, to fix it. My manager, Arie, writes to me that he’s not aware of the problem.

In the middle of the day we had a three-hour-long meeting: the guys were discussing something like ”private diagrams, and I have no clue what is it; so they were discussing it very energetically, and I was listening very patiently.

This is how I have spent my day. It reminds me something.

October 27, 2021

It would be stupid to complain.

First, I made sure it builds, but it finds errors in the code. The errors are okay, real errors, I’ll have to fix it. Two or three. So I fixed them.

But then Sebastian asked me, why am I using a version of maven that is too new. Really, wow. This version is too good.

Meanwhile, our devops decided eventually to start using https on their new artifactory.

Okay, I installed the six-years-old maven, version 3.3.9. Okay, everything still compile. I have stashed my fixes; the tests still pas. Isn’t it nice. And everything compiles. There were bad configs.

Okay, and at the meeting I reported my achievements; they gave me another task: add a new User Role to the system. Sure, I will. But let me first upgrade my MacOs, we are now in Monterey. So I'm sitting upgrading it.

Meanwhile it's too cold outside, so sitting there is pretty uncomfortable. At home I don't have any chairs yet, so I have to stand. What can I do? Bring a chair from outside? I'll think about it.

And I was bugging the devops, because, you know, I try to open Jira, and Jira switches me to "how can I help you". So, we talked, and talked, and they say: last time I logged in it was October 19th. Oh, really?! Ok then, let me log out and then log in. So I did. And yes, it got me to the right page. Isn't it beautiful. Really beautiful. Good job, Atlassian. There were days when I respected them a lot. I was setting up meetups in there office. The office is not that comfortable, in a dead end, but it's okay. OMG, Atlassian... е мой...

October 28, 2021

Well, today I had to give a talk, at "Scale by the Bay", so I did not work much. Just a little bit.

Finally I reached Jira, read what's there for me, and decided to add more permissions to a JSON file. Why not?

But then Bill came over to help me. We spent about an hour and a half, on Zoom; he was even recording it. Probably for the posterity. For the movie fans of the future.

He was trying to teach me to use IntelliJ. That's where my patience kind of ended, and I started, in a simulated New York talk style, explained to him that I've been using it over 15 years (but I didn't tell him that I personally know the authors). So, as a whole, the simple problem of adding roles with permissions into the system, grows into a huge epic with adding all possible permissions, and, what's more important, implementing them in the code. Because how else, if it's not being used in the code, what's the point?

So, in the end of the day I wrote to Arie a long message, asking "hey manager, what are we going to do?!" I could of course happily implement standard ACLs, but it would be good to have less hallucinations in the project. No, I didn't write this. I just suggested to plan our activities.

What I like in this hospice is that all the topics are pretty much familiar. I even use the terms that are, it turned out, standard in this company.

Okay, we will see.

October 29, 2021

Just heard at a meeting: "Santhosh, our DB tables are immutable!"

Wow, I can't believe it. I spent years and years that UPDATE is bad, bad, bad. Except for the case when we are fixing something.

Otherwise, meetings all day, and some code reviews. For the stuff I wrote to be a part of the server, I need to prepare pull requests in two modules. Bill was coming up with his ideas, but I promised to write a plan, so that all the ideas are implemented in three easy steps. The pull request for one module was immediately approved; as to the other, it turned out that I have to add a reference in this file, and in that file. Not a big deal, in general. Everything is just fine.

Except that I combined the meetings and the conference, the whole day. For instance, currently, at 9pm. I'm listening to Orendorf's talk, where he implemented linear algebra in Agda. Like I did it, a couple of years ago, in Scala. Well, it's not 100% agda. It's a mixture of Agda, Haskell, Python. But it's beautiful.

November 2021

November 01, 2021

The work day hasn't ended, but it's already dark outside.

In short, we discussed my design doc; my pull requests are approved and merged. Now they want me to urgently deploy. That's what I'm doing. All the scripts are in shell, and they are pretty cultured and logical. And I'm working on this deployment part.

tThat's it for today.

Well, now I'm still trying to deploy; for which I'm reading all kinds of README, and trying to attach aws credentials.

(Sigh) if somebody could slap together scripts for all that.

November 02, 2021

I was too busy today.

Okay, Sebastian is showing me which buttons to click. All the docs, all the READMEs are all obsolete.

In reality one has to log in to one server (Sebastian told me its ip address), and run there a script that I was running locally. It should be run on that server. After which, on Slack, give a bot a command "deploy the newest thing right now".

Then our Indian QA guys were happy - but they immediately discovered: one files with roles is missing. Sebastian did not know about this file. I had slapped together that file, pushed it too, and as soon as the build finishes the bot on Slack will report to a special communit, and I will tell the Indian guys: so-and-so, we can celebrate, the roles are set.

Does it look like software development. It looks like some crazy shit.

Well, whatever. When it all ends, I'll have to run the tests. In general, this company is a little bit more advanced than the previous one. What I was doing over there (pushing the test environment into Docker), is 100% ready here, you can run it ad libitum.

Kind of funny. Maybe I should invite Krzysztof.

November 03, 2021

Well, so, I worked in the morning. But, but... in the morning VPNs (the company has them four) didn't let me in, but Stephen, a good-hearted Vietnamese guy (I never met other kinds of Vientamese), fixed it real fast (I don't know in which time zone he is). Now everything's fine. I finished fixing all these

configs, filed a pull request, it was immediately approved, I merged it and deployed it. With no stupid bureaucracy of which I saw so much all these years.

Krzysztof writes that Sandeep sent him a “long letter”. The night of long letters. In response I asked him if he would like to try our company; he doesn’t mind; and I pinged Arie, and Arie also doesn’t mind. Very good. I’ll organize it.

We had a meeting at 1pm; they gave me one more Jira case, and regarding the one that I have supposedly finished, but deployed to Dev only, the Indian guys told me they’ll catch up with me as soon as they finish testing it. Seems like we have QA engineers. That’s unusual.

I know, I understand, it’s a honeymoon, it lasts usually two months in a new company. Like in “The Voyage Out” by Virginia Woolf: *“I spent six weeks on my honeymoon in having typhoid at Venice,” she continued. “But even so, I look back upon them as some of the happiest weeks in my life.”* (quoting by chatgpt, but I rememberer the quote because I read the book)

Before switching to the new case, I sent a pull request, where, in README, I put all my acquired priceless experience, “for the noobs”. When you write something “for the noobs”, the proud veterans ignore it, instead of looking for typos or for principal design errors.

Oh, and I also wrote, during the weekend, a text regarding RBAC (role-based access control), and somehow they are reading it now, attentively.

I’m waiting for them to start blocking my work. Will they?

November 04, 2021

I’m gradually switching to the evening lifestlyle. But today I had to get up at 8am, because they had to bring wine from NakedWine, a nice company, I don’t know how they make ends meet; I just, like a total loser, filled in a questionnaire on twitter (my answers were correct, well, I’m a bit of a connoisseur), and they sent me six bottles of pretty decent twine, for \$35. Would you fall for such a trick? Normal people don’t. And so, yesterday a UPS came over, but they did not leave the wine, because it should be delivered from hand to hand, and I was walking at that moment. So today I was sitting on a chair in front of the window and watching whether they’ll deliver or not. They did. Well, and after that I was kind of working (there was a new case, I had to add a field to a form... not exactly, but almost like that).

Okay, I did it. And right now I’m compiling, because in Bill’s code there are var, var, var all around, and also tons of nulls, and also DI, so, in short, all this must be fixed, and for that, probably, Bill needs a prescription of haloperidol.

Meanwhile, Krzysztof from Video2 writes that he’s exchanging long messages with Sandeep (Krzysztof needs to get out of there without returning his hiring bonus); Krzysztof writes to Sandeep that we,

proud Polish guys, never lie, and we don't accept the lies, and if you want to talk to me, stop lying. And Sandeep responds: You, Krzysztof, are sending for a review a pull request that's too complicated, and you merged it while Chiyu did not approve it, but wrote comments. Generally speaking, what's his, Sandeep's, business? But Sandeep is afraid of Chang, and Chang supports his Chinese friends.

So I got Krzysztof in touch with Arie, and let them find a solution. It's great to work with Krzysztof (and one aspect of it is really, the Polish principle of never lying). By the way, Krzysztof tried to interview to that company where Kevin and Lea ran away from WeSell. A proud Chinese girl was interviewing him, asking dumb programming questions, and in the end Krzysztof asked her a question: can you write a derivative of e to the power of x. The Chinese girl was totally pissed off and said: "who interviews whom here?!"

He's the kind of person with whom I'd work again.

At 1pm we had a standup (it's a permanent standup for me: I brought a chair from the porch, but I sit on it only in the farthest room, but I do the rest while standing. Everything's fine. They asked me: anything blocking? No, - I said, - everything's fine. (I'm mostly trying at the moment to keep a decent order of parameters of the constructor.)

So, what do we have. It does not compile for me (the problems are dependency injections, vars, and the other shit from ancient mammoths I'll fix it, no problem).

Then I had to go do my chores; when I returned, I worked again, but first, I wrote to Jordan, saying: hey, talking about runType, you made it an enum; maybe we join our efforts, I was going to use it too. Hmm, Jordan is not aware that he's making an enum, and he sent me to Ben. I went to Ben, we turned on Zoom, discussed it casually, and agreed on everything within 10 minutes. So far we will not do a too strict typing. String typing would be enough for us for now, and we'll see later. Because we have a frontend, we have a backend, and they don't talk to each other by strict typing. You may disagree with me, but that's my pretty established opinion.

That's it for today. Everything will be fine, I hope. Because the market does not feel well.

November 05, 2021

Meetings all day, starting at their 9AM.

But I actually started working even earlier. Bill has slapped together tons of merges, and my code became incompatible, from the p.o.v. of github. Big deal, I've slapped together a new pull request. Frankly, the new update from Bill made the code a little bit cleaner, updating it from the style of year 2008 to the style of year 2010. (sigh). Whatever. I'll work on it.

But nobody's reviewing it.

And I sent another pull request: in one project I've edited its README, mainly kicking out the garbage and adding a reference to wiki pages.

Then the meeting became a) boring, b) overlapping. I didn't always get to the right one. Around 6pm (our time) I just closed my work computer, and went to swim and walk. Well, swimming was brr, too cold.

November 08, 2021

And is there anything today? Not much.

Nobody's reviewing my pull request, and Arie asks me whether he should bug somebody. Yes, please.

Also, I need to deploy my shit to Staging, and I'm trying to get a confirmation from the devops, should I follow this scenario? Yes? YES?! Hello!!!

Also, I got another Jira case; not that they threw it to me, it was me who begged for a case.

That's how I live these lazy days. All I do is I read the docs.

November 09, 2021

Nothing today. Well... this and that was fixed; something was deployed; the config file wasn't deployed, and as a result, the feature isn't visible. We managed to convince Eric to deploy the config file too. But then they told me that the config file must also have a permission: and I added the permission. I don't know what ERic is deploying; roughly speaking, I don't care much. I'm studying the app, the code, and one, kind of nonsensical, recommended lecture on ML, from Google, where the speaker seems to not know the difference between a multiplication of numbers and a Cartesian product. I know, this is how the world works. Also, the market fell by the lunch, and went even worse later on. Good that I sold ABNB yesterday, at least some gain.

November 10, 2021

So, it's almost a month that I've been with this company. Pushing features in production, philosophizing, fixing stuff, writing design docs. So far nobody's fighting against me. How come? What's wrong or something? Where am I working? I already forgot how this can be that nobody's fighting against me.

Now, what's going on? I was setting up the VPN, because the previous setup was wrong; I was pushing a feature to production; there are config files to fix. But it's almost 9pm here (it's 6pm there). As soon as they approve it, I'll merge it, and will tell devops that they can deploy it. They don't allow me to

deploy to Production yet, saying that I didn't pass a HIPAA training. I'm actually sure that I did; will need to check whether I did. And it seems like I'm lucky this time (no, you are not).

November 11, 2021

Spent all day trying to merge my approved pull request to master. What is it? Nobody knows. Github. In short, in the end I've made another, a new pull request; now I'll compare it with the old one, will write a good comment for it, and I'll merge it. It's annoing. And then I'll try to deploy it.

November 12, 2021

This morning I suddenly decided to start working, right away. I cut my pull request into smaller pieces, and merged them right away. How come? See, a) code review is not obligatory at this company, and b) does anybody run the tests? Who knows. I do run the tests locally, but github just doesn't care: go ahead, deploy it. So I did deploy it. And at the meeting I reported the deployment. And see, this is the day of end-to-end meetings.

Now, what happened with the merge? Ha, Bill was merging simultaneously. And he had the same shit. Why? Because he was trying to convince me to synchronize when merging (the same file). Ok, sure. Why doesn't he synchronize?

So, you see, the same ancient game, rock-scissors-paper.

At the last of these meetings I got a new case: write a design doc for this and that. Sure, I'll do it with pleasure, but first I have to investigate, what exactly does this UI need. But, in general, it's nothing.

And in the evening, when it was already dark here, Bill sent me a pull request where he rolls back a bunch of my updates. And I write to him, asking: what's up? Something's wrong?

No, I understand. But I'm supposed to not care. Did I subscribe to fix all these hundreds of nulls, 20-character identifiers, methods taking 30 parameters, 2500 characters long classes? Bill is an ancient man. I saw such people God knows how many times. He might have learned some Object-Oriented Programming, but Functional Programming for him is devil's tricks. No, at times I see map in his code, but mostly it's just var, var, var. Am I supposed to reeducate him? Cure him? No, it's not my business. But signing off the code with which I disagree, that's not what I do. You can merge without my signature.

At least the market goes in the right direction today. Yesterday, when it was down, I bought more of the stocks that fell through; and today I sold that part of them. It's okay, everything's fine. But I'm about one months back. Isn't it horrible.

November 15, 2021

I had to write a design doc first of all, but I was busy writing code.

By the evening I wrote a script that, if you kick it from your laptop, it prepares a Dev deployment. I could also do the deployment itself, but there's a Slack bot for that. Then I sent this script to our "system people" (that is, our devops).

Мне бы надо дизайн-док навалить, но я увлекся программированием.

And I was right not to write the design doc: we had a meeting around noon, and we also talked about this topic; at least I heard it mentioned. Now I can finish writing the doc.

Two other stranger things happened. The pull request that Bill, a founder and a stylist, slapped together, I asked a couple of questions about it, in the style of "are you really sure?", Bill has thrown it away. Wow, wow. I think I was just carefully expressing my vague doubts.

Also, we have some guy called pg, he's some kind of a VIP (it turned out we are talking about Prasanna, the CEO). Bill mentioned him, sayin that, see, the identifiers, even if they are used in JSON as strings, must be written in *camel case* (I don't care, I just copied the style from the previous line), and I asked some question, this and that... and he had answered *vaguely*, saying that, yes, we have such plans, but this parameter, it's, for now, optional... Now I feel it's not the practice I'm used to, where everyone has a strict opinion and tries to enforce it. What can it even mean? I don't know. Why should I? I'll try to be friendly. Not that I'm a master of a friendly communication, but I'm now a Southerner. Everybody's armed around here.

November 16, 2021

I didn't work much today.

Nobody touched my last night's pull request (I'll have to ping them). Then I wrote a new one, after talking with Prasanna. And I'll have to work on the new case.

There's something weird there. We have tons of code, but where's the class that I need, Case? Ha, it doesn't exist. We only have one class, Entity. Interesting, I'll investigate this approach. I will. And now I have two cases in Jira, so there.

There was a meeting. I mostly slept there. Meaning, I was listening, but didn't interfere. I'll have to investigate this data structure. Then I asked Arie, and he showed me where to look. Okay.

That's it for today.

November 17, 2021

How kind are the people here, in this company. As soon as I said that I don't see, where we have records and where we have classes matching the records, where do we have requests and responses, Bill just opened Zoom and showed it to me, and that was it. What's funny here, we have Entity everywhere. No specialized types. Just Entity, with fields. Purely a dictionary, like in JavaScript or in Python. Well, okay. We can live with it. Pure lambda, with Hibernate.

Now it suddenly turned out that the case on which I started working just disappeared from Jira. But there is another case. And the data there, I could not even find, where is this entity! Now Bill and Jordan together showed me a page where all this is present. Wow, okay, thank you!

But the thing is, a combination of two fields is being used as an id. And Sridhar wants to modify these fields. Wow, but this entity (with this id) is referred by other entities. This is funny. OTOH, it's good that I don't care anymore. Absolutely. It's impossible to fix it, because, to fix, one needs a party. A team of like-minded people. And such a party doesn't exist. Well, Jordan is okay. And if Krzysztof come over... if Krzysztof interviewed today to this company. He liked it.

That's what's going on now.

When I asked Sridhar, what are we supposed to do with deleting the records that are referred from other records (I don't want to tell him about foreign keys), then Sridhar wrote that he'll talk to the team - and went quiet. Well, of course, it's already 5pm on his side, too late for doing anything. So I turned off my computer for today.

Just stay quiet.

Meanwhile, recruiters from a variety of companies are contacting me. Some even talk about 700k. I actually don't believe them. And, probably, it makes sense to wait (no, it didn't). I wrote to that recruiter: let's wait a little bit, I can't leave right now.

November 18, 2021

Finally, I've figured out how to make that shit that I have to make.

What's not there. Types. There are no types, that's in the database too. It's familiar to me. In around 1995 I had invented and introduced, while working at Tangram, a flexible database, which, basically, is known as "Entities with Properties". Like dictionaries in JavaScript or Python. When the owner kicked me out of Tangram, the people had immediately forgotten who introduced this database and started calling it after Aristov, the guy that was given to them as my replacement.

In the other news, a new guy joined the company, his last name is almost "Carl", but with a bunch of unpronounceable 'h' - something like Kahrlh. One would expect this kind of word to come from Ireland, but he's from Mexico. He lives in Washington (that is, Seattle), and he rides his bicycle there

and grows his three teenager kids. He spent 8 years at Amazon. My condolences. Bill writes to him: “welcome abort”. FIIK what this phrase can mean in our professional jargon.

At the meeting Bill gave an oath to fix his bug during the weekend (I’d qualify this bug as belonging to both of us: Bill convinced me that Prasanna doesn’t like spaces in identifiers, and I said: okay, let’s fix it, and he said: I’ve already fixed it, and then what, then I approved it. But the problem is, okay, you’ve fixed the identifier, but how will the frontend know about it. And besides, to merge a cherry-pick with this from Staging to Production, we need to browse through 94 pull requests.

In short, I’m not exactly interested in curing all these crazy people, even if I had a lot of ivermectin and haloperidol.

Now someone else is asking whether there is something in Scala that would do Web Sockets. Then I remembered that a while ago I did play with it, had no fun, and quit. But there is ZIO, and it has everything. So I sent him a link. Myself, I plan to start using ZIO as soon as I finish editing my book (pretty tired of it).

In short, it’s a little bit of a mystery. But as long as they don’t prevent me from doing the things I find necessary to be done, I’m fine. Isn’t it a challenge, to fix everything so it would just start functioning by itself.

Krzysztof is interviewing, this is good. Although I already have some caveats about this. Just a little.

Also Krzysztof is curious how to make it so that they kick him out without demanding his hiring bonus back. I suggested to use more sarcasm (for which I was kicked out from that company), but it’s complicated for him, he’s a serious romantic guy, the type of Chopin. Or work at both companies at the same time, and let Sandeep, or what was his name, goes crazy, what’s our business. Let him go fuck himself.

Okay, meanwhile, I’ll go continue working. It’s interesting! It’s a challenge!

November 19, 2021

In the morning I started writing the code, in the standard style, API/JSON/POSOS/Postgres; it’s all traditional, but a lot to write.

Then we had a meeting, and, to my amazement, I’ve learned that a) data structures are defined on the client side (they do have schemas, but the schemas may change), and Scala may have no clue, what’s inside, and the server probably shouldn’t even know. In short, we just need a thin layer between the Frontend and the data base. Now there’s a question: guys, how many clients do you have? Just one? Or every web page builds its own data schema? Or? In short, curioser and curioser. But, on the other hand, in reality the schema is also stored in the database. I’ll have to investigate it.

After the meeting I started nagging Bill, asking: really, we don't need to be interested in what's inside the object? And he answers: no, otherwise we'll be storing a string in a date field, and a date in a number field. And so it goes. Wow. Oops.

In short, I'll have to follow their example and write all those annotations. Which, of course, makes a little bit of sense, but just a little bit. Okay...

We have a day of meetings today (6 hours of meetings, with a lunch break; strangely we didn't sing any anthems during this really long day).

Okay, and they threw me another Jira case, and I, while sitting in a meeting, was implementing it. But then Jordan told me: no, the field should not be a part of this class - well, okay, I've moved it to another class, big deal.

But by that time most of the participants already quit the meetings, to do something at home, especially Bill, who lives somewhere in upstate New York.

Another grandfather with a ponytail was hired. I got a feeling that I may be one of the youngest in this company. No, actually they have about three engineers younger than me. The vague general idea is: we hire the smartest ones. So there.

So there. I guess I should not hurry with promoting "web applications standards", it's something a little bit trickier, and I'll have to learn more. But my suggestions regarding "let's refactor it a little bit" are accepted with an enthusiasm. Weird. I'm not used to it. Of course, it's a honeymoon. I'll enjoy it while it lasts.

November 22, 2021

I had two pull requests pending review, one about a deployment script (why would I copy-paste commands from wiki if I can put them in a script?), and another, adding fields to the "form" (actually, to a class). The first one, we are still discussing it. As to the second one, Jordan has approved it, but he wanted Bill to see it; but Bill doesn't respond.

The buzz at the scrum meeting was pretty friendly. When it got to my small feature, I told them that it's merged now and is building. I didn't tell them that I've been watching a movie while it's building. But since I have a script, my manual actions are reduced to a minimum.

During the meeting Arie asked me, on Slack, what do I think about splitting our huge service into two. Wow, I've been always a big enthusiast regarding microservices. More, since these two code bases have two very distinct life cycles, then of course, let's do it. I opened the Jira case, wrote a plan of actions (writing is my hobby), and wrote to Arie that Christmas time would be the best time to deploy the new version, why our customers are away celebrating. And here Bill suddenly pops up and writes: hey, I've already cut it into services.

He didn't exactly cut anything, he just moved one module to another folder, so now he can build a separate jar file. And he writes to me, saying: let's talk on Zoom. I respond: sure, but let me have my lunch first (It's already 3:30 pm here, and he's in the same time zone). Silence. So I had my lunch, and at 4pm I write him again: Bill, where are we with Zoom? Silence. I went to walk my miles, my thousands of steps. When I returned, I see a message from him: let's talk.

And we turned on Zoom, and discussed it. Since he knows nothing about the deployment, and he has no time for it: he has contractors at home, today and tomorrow, and the day after tomorrow, we agreed that I'll talk to senior devops, then I write the docs, and inform everyone potentially involved. And by the way, since this is a short week, I probably won't find any devops, but I'll try. We can do it in steps.

I don't understand yet, who, do they think, I am here. But I like it. It does not seem like they view me as a dumb laowai. I'm trying to do my best. Writing docs is the easiest part for me.

Also, I got an information from the devops,, where I can find the logs. So, cool, celebrations! I'll investigate. I'm really curious about what's in the logs. I think they are full of NPE. Btw, today, in the morning, the devops were investigating a StackOverflowException in a servlet. How can it be? Is it calling itself? And all those hibernates, which professionals hate. We will see. It would be great to get rid of it all.

And so this is how the work day has passed.

November 23, 2021

Bill is busy with contractors at his new house. Those devops whom I asked to get together and discuss things, they just invited me to their meeting on Wednesday; and I'm still discussing my deployment script with other devops; Arie is asking me when my shit will be read (a new class and its API; I told him: by Monday); and that's it. I worked in the evening, and still working now. I'm on the evening shift, so there.

November 24, 2021

You know, if there are no horrors at work, there's nothing to discuss then.

But, so far... for some reason, my pull request fails, in stylechecker, and I don't even know how to run the style checker on my tests code. I knew, but then I forgot, and forgot tu put into scripts. There's a link that Google returns, but that link does not lead anywhere. So I have to just gess: commit, and then see whether it fails

At the 1pm scrum meeting there were just me and Arie; we talked about life, about Carolina, about new Scala engineers coming soon. Good news.

Another meeting was at 3pm; they invited me, but I missed it, and joined only at 3:22. But it's okay, they had things to talk about. I had questions, like: hey, how are we going to separate our module and deploy it separately, what technologies do we have for that. Prasanna and Eric (a Chinese guy) started pushing what they know - but shit, I don't understand what they are saying! Approximately, I understood that builds a made weekly, and everything seems to be pretty clean. Then I have a question: why do we have to do something, let's continue with what we have, a monorepo. Now Prasanna, whose idea it was, explained it, that our module is in "alpha state", and it should be deployed more often than once a week. Well, okay. I wrote down all these ideas, grabbed a picture from the screen (it had a useful url), and stopped participating in discussions.

I'm still fixing all this shit about "style in tests", omg. I tried to turn on the style checker for all the tests. There are over 60 style failures. Oops. I don't have time for fixing all their tests.

November 25, 2021

It's a weekend today, but I'm too lazy to do nothing, so I worked. What I did was upgrading the deployment script, regarding which the devops were complaining, and also started implementing this new type and a CRUD API for it. Our structure is pretty dump: its Hibernate; I did refactor it a little bit, but I don't want to scare the people, and also I don't want to get distracted... they won't get it anyway.

Ok, here: I have coded one kind of CRUD requests, seems okay; will have to have a test case for it.

No, it was step 2; I couldn't squash step 2 (too complicated for git), so I wrote a new one, a clean pull request with just one commit. There are problems with squash. Who needs it, no clue. Sebastian? I don't know, maybe I should talk to him.

November 26, 2021

Spent almost all day working, inserting CRUD operations into this shit that they wrote. Trying to figure out how to write decent code in their framework. Okay, I wrote it, let's see how it will work. No tests yet, just assumptions. They, as usual, have about five times more code than any common sense would have required.

November 27, 2021

It's a weekend today, generally speaking. But I promised to them to finish it by Monday. In short, I've slapped together the code, now I have to swap together the tests. This will be the third pull request. That's it for today, and tomorrow I'll work on tests.

November 28, 2021

I didn't work much today, but, otoh, I managed to slap together one end-to-end test, but now it's not clear how they expect me to run it.

November 29, 2021

Just to mark this day.

I have merged one pull request, and as to the second one, they asked me not to merge anything until tomorrow, they are doing a release. Do I mind? I do not.

They've just attached some Codacity to the system, and now it writes all kinds of stupidities about the code. And nobody knows where are its config files. On the other hand, one can ignore it for now. Since if you follow its demands, the code won't compile. So there.

Not that suffer. Instead, I signed up for yet another Jira case (although I have not finished this one, and didn't start the other one).

But, eh, I doing all kind of stupid shit, but where are the tests?

Meanwhile, Krzysztof writes that Chang is quitting (that, in "Video2", my previous workplace). Wow, how is it? He lost the fight? Is China losing the fight? To whom? Will Sandeep replace him? Hardly. Not that I'm interested. Just a vague curiosity.

November 30, 2021

It's kind of funny. I merged my stuff, and an hour later Sebastian is writing to me, saying, hey, our end-to-end tests are crashing after your merge. I answer: let me check. I go to Jenkins: it does not let me log in. This way, that way - no. So I wrote to the devops. Silence. Ok, I'm waiting. Meanwhile, I'm trying to run the tests locally. And I see, in one place, Jenkins login and password, right in the code. Oh, nice, thank you, Sebastian. Now I log in, to check, what has crashed. His tests crashed, in kind of murky circumstances ("wrong sql statement"). I wrote to him. And asked a question. He responds: "I'm very busy now, can't answer; ask the Slack channel." This is funny. Okay, I asked. They showed me the ropes. Does it help now? Now what they show, it's all garbage, and obsoleted docs. Ok now... I'll figure it out. Right now, no mood for it.

By the way, our tests, they are totally mental. You launch a test, and it sends a request to Dev, and checks the response. What are we testing? What kind of garbage is it? Well, and besides, most of the tests are annotated as disabled. Meaning, we have no time for tests.

And now they crash.

And Bill writes too, like, here, I run some command (something like mvn --disable-tests=true...), and it crashes. Wow, interesting.

And at a meeting, two, I'd say, idiots, require from me that I write a CRUD, so that it would send, via WebSocket, from server to client, the status of the operation, progress status, that is. Meanwhile, the operation on the servers is distributed. Well, we could try to calculate the average status, averaging the statuses of individual operations. But how is it related to CRUD? I tried to get from them, at the meeting, do they want, in the process of status reporting, to delete the data, or create new data. They didn't understand my question. They need CRUD. So that the progress is being reported. I think it's beautiful. (And now, in 2026, I think it's feasible, but not in the bedlam that this company was (or is?)

December 2021

December 01, 2021

I was busy with a bunch of insignificant, small things; especially begging the colleagues and the devops tell me, how come I don't have access to the artifactory, and the jars are not available (and there's nobody to ask why I even need it).

Meanwhile, I'm having a vague feeling that there's nobody else here to write code, and that they throw at me more and more tasks, one after another; and do I refuse? I don't. But let's make sure our infrastructure makes some sense, and less psychiatry. I can write the code, it's not a problem, but if you are competent, answer my infrastructural questions.

And there's an eternal discussion regarding CRUD API for streaming the status via WebSockets. Where's CRUD there? Its a status stream. I don't exclude it that I managed to convince them. Or at least to turn this Titanic the right way, away from the iceberg.

December 02, 2021

Nothing to complain about today. With the help of Arie I got a hold of Stephen, with the help of Stephen I logged in to AWS, and from there I have downloaded an image for Docker. Now I'll be running it. Well, first I have to figure out what's inside. Because... well, it was me who wrote the instructions.

Then I merged one small pull request; nobody was reviewing it, but it's nothing, it's just a tests cleanup. Well, and then I wrote another one, on the same topic.

And there's another pull request, it's waiting for a review, but now it's funny, now mvn clean test is crashing with some nonsense inside JVM, and the authors of these failures are blaming me, because I was the first one to discover it, because nobody else runs no tests.

They seem to be ready to hire Krzysztof; I hope he'll agree. God willing. We'll rule it.

That's it for today. I have a feeling that I work the whole day, 9 to 9. I started recording my screen time. And the screen time shows that I don't work much at all. Hmm, funny.

December 03, 2021

I continue wondering.

Okay, here are my pull request, I sent them at my 9am, their 6am. Bill wrote later, commented something. I don't mind.

At one of the meetings they are telling me that what we need is not CRUD, but CRLDU: add getting a list, with filters. NP, actually. They also want to specify Business Unit not in JSON, but as a part of the url. No biggie.

What else was happening. I'm still bugging them about hey, guys, HOW CAN I RUN INTEGRATION TESTS locally? And all I hear is some vague nonsense. Chinese geniuses.

At 1pm we had an “all-hands meeting”, where Prasanna, for two hours, was talking about the necessity of a good test coverage. That's what we were talking about with Arie, yesterday. Wow, the management is getting it. Gradually. (Now, in 2026, I think that it was like this: I told Arie, Arie told Prasanna, Prasanna told the team.) I was holding myself back, to avoid coming up with advices. It's a good thing, when you can just ignore it.

And that was it for today. Will check again my work computer, and, maybe, merge something. As to the integration tests, the Chinese geniuses are now writing me that I can steal one special file from the Dev machine, and use it. Ha, wow, I'll steal it. And I'll document it. And put it on github. Guys, let's keep in secret the PII and the real secrets, but the rest, the test data, and fuck off about the rest of information. Well, that's a theory, Business is everywhere like this: monkey see, monkey do. What's the point of understanding anything, for that, one needs to understand. They don't pay for understanding. We will see. Life is beautiful. Except for the market that kind of crashed today. No biggie, I managed to buy something at this beautiful moment. And we will see about it.

December 04, 2021

Aha, I worked today. Created one more CRUD: “give me a list, by a filter”. I even sent a pull request.

But. Nothing works locally. Either it doesn't compile: this jar is missing, that jar is missing. And now Eric writes: my merge broke the build. Wait, how did it break the build? A test is failing. Which test? Why wasn't it failing a) locally, b) in my branch in Jenkins?

In short, it's a FUBAR situation. And the idea, by our Chinese geniuses, that we have to save on tests. Although Prasanna was teaching them, just yesterday, about the tests.

So, what happened they have a so-called sanity test. It's a good test, but how come we don't run it by default. Eric has fixed this test, and everything's fine. But the problem was, I introduced traits, and nobody did it before, and their test got confused. Nice!

And yes, I still can't run integration tests: I still don't have a file “with credentials”, and Sebastian is trying to convince me that this file is not necessary. Well, I can comment out checking the existence of this file, but then there will be a problem with Postgres. But why, why isn't it required? I'm curious.

This was the end of my sufferings for today. I have left them a bunch of messages. But they, of course, will forget these messages by Monday. What can I do, create Jira cases, or what?

December 05, 2021

Imagine, I didn't even open my work computer today. Weird, eh. I only spent time thinking. I'll need to make a fuss today, on the topic "let somebody tell us how to run integration tests, eh? Fuck..."

December 06, 2021

I wasn't lazy enough in the morning to concoct a pull request, but it turns out that github again does not want to merge it. Big deal, I wrote a new one, it's easier for me than cherry-picking. Sent it for review to Bill, and to someone else too. Bill did approve it, but he asked me to first deploy it to Dev. Kind of strange, right? Don't we deploy to Dev from master? But Bill says that one can set up a local Dev. Wow, how is it? And he said he'll show it to me.

After which he found yet another doc page with instructions, how to run integration tests locally; the trick includes logging in to a server from a browser and pulling out a session key from some cookies. Wow how hardworking are our programmers. Well, I could, of course, start using Selenium, but why?! I just want to run tests, that's it.

BTW, I'm running Postgres in Docker, but the servers can't reach it, for some reason.

This clinic needs psychiatrists.

And then, suddenly, Arie writes, asking whether I'm interested in joining a meeting. With pleasure, but it wasn't in my calendar. Ok, so I joined it. It's an hour-long scrum, which actually took 45 minutes, and we had the remaining 15 minutes to talk with Bill. Well, Bill was working with his tongue, and me with my ears. He's teaching me to look up cookies in a browser, not reacting to my comments that I know how and where to look for cookies in a browser.

Fuck, whatever. Then another meeting started; the management (pretty advanced and educated) and the PMs spent an hour and a half bragging about the features. Unfortunately, I don't have my headset, so this Indian accent (which is as ugly as my Russian accent) reflects from the walls of this empty room (all our rooms are presently empty), and it's horrible. In the end I managed to connect a headset (it wasn't easy, the headset attempted, all the time, to connect to another computer and to my phone).

And now Arie writes to me whether I saw a message on Slack. Now, what is it? Ha, Eric writes that Swagger doesn't accept my interfaces, because the Scala code that I'm writing it's too complicated. Wow, I know, I write in idiomatic Scala, and Swagger probably needs an idiotic version. Well, okay, let me check. And, Eric, where are the test that check what's generated? Can you give me the tests, so that I could check it myself, before merging to master. To which Eric answers: if you are so smart, write the tests: build a war file and run the web server. Wait, what the fuck? Why do I need a web server to verify a list of class files? Eh? But Swagger, for some reason, wants request headers. And cookies. Omg. Who writes code like this, guys... Have you heard of Demeter law? Didn't Josh Kerievsky write about it in his books? Did you read any books. Fuck. Whatever.

So I wrote a new version of that code, sent it to Eric, and he didn't even review it, but wrote: "go ahead, merge it". So, okay, I did merge it. Also I wrote to him that right now I'm really busy and I really don't have time for writing tests (that's a Chinese form of saying GFY: Eric is Chinese). Then I deployed it, why not. Let him run his secret tests.

In the evening I tried the trick that Bill suggested (including pulling cookies out of the browser). Ok, now I have the cookies. Somehow they are required to log in to AWS. Can't we run tests without manually extracting cookies, eh? (sigh). If I were more proactive, I'd have been screaming at the meetings. But now I just smile and repeat learning Hiragana, again. Well, why not?

That's how I spent my day. A little bit nervous. Especially when, at a meeting, this Bill (who's older than me) shows, via Zoom, a shared screen, where I can't read anything without a microscope, because the letters are about half a millimeter in size. And you know what I did? I went to the pharmacy and bought new glasses, much stronger, just to read Bill's screen. Yes, I was pretty pissed off. But why? What's the point? Did I lose money on the glasses? I earned a lot on the market today. Of course, this month I lost a lot, but it was before, and I know I'll regain it. Be calm. It's important. Everything will be okay, as the gangster were tellin Mickey Blue Eyes.

December 08, 2021

Yesterday I hardly did anything at work; but today I do. I only skipped a meeting at 10, because a notary came with the paperwork.

What they want from me, it's an API where there must be a new request, /list, as we agreed with Ben yesterday. So I wrote it, and sent for a review, right during the meeting. Now it's 6pm something, and all I see in the comments is: "if a class has a method isValid, does it mean that this value will get picked up by Json during serialization?" - and then silence.

Meanwhile, Bills' transformer broke exactly at the point that I was supposed to have "improved". Well, of course, there are no tests. I wrote to him that I'll fix it ASAP. Wrote a fix, wrote a test, found more bugs in his code, fixed them, sent him a pull request. But now what. Bill was not wasting his time. He slapped together his fix (my test reports it as an error), and, meanwhile, he "canceled the traits", because they "make the code more complicated", and inserted into a huge class (5000 lines) two my traits. I wrote to him a small epistola regarding Good Programming Practices. But fuuuuck...

Aha, and also they asked me to move one API parameter to path. Well, ok, I took a look. Hey guys, will this parameter always be present? We are changing the format of URL? How about the people on the client side, did they hear about this change? So I sent them all a message, and that's when my work activity ended today. What can we do? On one hand, they want it badly, on the other hand, by 4:30 they have all gone to bed, or whatever.

Okay, we will see. I still can't run integration tests. I'm kind of getting close. Actually, right now no

tests pass. If you write mvn clean test, the majority of the tests are failing. And nobody cares. Do I have special needs, like to have tests pass. Of course not. It's probably just the taste that I think I have.

December 09, 2021

Bill has shown me a new trick. He wrote a pull request, where, since he's afraid of Scala traits, he stuffed all my code, from my pull request, which he was supposed to review, into one spaghetti bowl, which he already had.

I wrote to him, politely, that this is not the right way to do things. My pull request is still waiting for his review.

Now what he did: he wrote a new pull request, identical to the previous one, and merged it. Profit!

As to the rest of the story... it was an active day today.

In the morning I figured eventually, that it makes no sense to argue with Bill, and if he's afraid of Scala traits, and so he threw them away, almost all of them, and since he doesn't understand what does it mean, type Request = SomeRequestType, defined inside a class, there's no point to try to educate him, until I hold all these tests in my hands, the tests about which he has all kinds of myths in his head: so I'll wait for a while, then inject my code, and then will try to master the tools - that's what I did, and wrote a pull request for that. By the way, we had a talk over Zoom with him, and almost agreed on everything, except that this old geezer got obn my nerves, talking slowly, impressively (he was a professor a while ago): but how old is he?! 80? 65? Ha, I've just googled him. Seems like he's about 55. Almost like my daughter. So weird!

At the meeting they were asking me, "when will it be ready", and I said: here, the pull request is ready, I'm waiting for an approval, and I sent it to Ben, because Eric and Bill are not interested in other people's code, but Ben needs it fast. Ben seems to be the most reasonable here, in this company. He had approved it right away, and I had merged it right away.

And then I started thinking: how come Bill had merged his shit, who approved it for him, if I wrote to him that it needs more work? So I traced it on github. Ha, I see. He slaps together a pull request, and he merges it right away, not worrying about any reviews and approvals. Because it's legal in this company. And, it turns out, he slaps together about half a dozen of pull requests, and merges all of them. Isn't he creative!

He ignored my message that his code does not pass a test I wrote. Well, what can I do. I'll fix it later. Ha, I have a free moment now, let me do it right away.

Well, almost all tests are failing now, and I found where they fail. There is before class has a method called copyCredentials, which a) doesn't work (it tries to copy a file out of a jar file), and, if the

file is found, it skips the file, without any checks, and further down it “initializes access control”. Why is it called, I’ve no clue, but I have my suspicions. This “access control” wants Postgres to be running (Postgres credentials are hard-coded in its code).

So I wrote to Eric: hey, do we have it documented that unit tests require a running Postgres? He answers with a question (I don’t think he’s Jewish, though): “did you merge the last master?”, “did you read the docs, the link to which I sent you”? I sent him, in response, four links to four distinct documentation pages, where everything is different, four versions of docs.

He doesn’t answer my question. Instead, he writes, to a “developers channel”, saying: Bill, help Juan, he does not know how to run unit tests on his machine. I got pretty much pissed off. I wrote there that his code crashes in the tests, in this specific place. And whether he thinks it’s reasonable that a function that just copies a file, written by him, also initializes the access control service. He answers: “It was not me who wrote it”. I traced the history, gave him the date when he wrote it: in 2015. After which he shut up and stopped responding. Now Bill interferes, and says that we should not play `git blame` here. And now another Chinese genius pops up, asking me whether I copied this specific project to this specific place, as specified in the docs. Ha, sure.

There’s more fun actually. If you follow their docs, and their follow-up advices, nothing will work. But I’ll investigate all this tomorrow, in details. Instead, I wrote a pull request, where I fixed that function, and I also started checking port 5432, to see whether Postgres is running. In general, it is not a great idea, but this function is a part of test, and, as Josh Kerievsky teaches, everything is allowed in tests.

I slapped together a pull request, sent it to Eric, and then I started to investigate, how come his exception reports line 100+something, if I only have 80 lines in this file. Checked out, where does this file come from. It’s in maven cache, where the jar file hides, it’s a SNAPSHOT. So I renamed my `~/ .m2` to something like `bak.m2`, and ran the build again. Maven, sure, started pulling all the dependencies again. It turned out that the snapshot I have is too old, and the new one does not get extracted from artifactory: I get a 404. Very, very nice. I went there by its URL. Still getting a 404.

I wrote to those idiots, where is the file you say you have in artifactory? Maven’s pom file wants this snapshot. Silence.

So I had finished my daily toils and troubles, closed my work computer, and went for a walk. I’ll think about it tomorrow. ï

December 10, 2021

As I see, nobody’s reviewing my small pull request, where I separated two functions in a test tool, so I sent it to David, the usual victim. He wrote a lot of code here, so, maybe, he’ll appreciate it. Meanwhile, it’s Friday, the day of meetings. At the first meeting Prasanna was giving a beautiful speech about our tool and our UI; then there were discussions on how we do this, and how we do that, and here we’d

need such and such data type, that's one thing (here I suddenly came up with an endorsement, and also suggested to introduce a class `Sentinel`), in short, if someone comes up with a meaningful suggestion, and somebody else supports it, there's a progress, and the PMs will add cases in Jira.

And then suddenly they started discussing integration tests: it turned out, everybody needs them. Bill, on the other hand, does not understand, what is it, an integration test (and I already wrote them a document, explaining what is what, and where). Well, what can we do. He can learn.

Then Eric asks me, why do I need a jar file from the artifactory. Fuck, no, I don't need no jar files from no artifactories, I need classes, and I'd prefer to get them from the module, not dragging around the compile ones. And no, it seems, Bill has broken the project again. He did it carefully: it builds on Jenkins, but not locally.

Talking about Jenkins.

In the evening I started literally, step by step, do what Eric's doc says. At step 5 it doesn't work and it won't work:

```
~/projects/private/pmmp/book-modern-math>ln -s manuscript /opt/share/man  
~/projects/private/pmmp/book-modern-math>cd /opt/share/man  
-bash: cd: /opt/share/man: No such file or directory  
~/projects/private/pmmp/book-modern-math>ls -l /opt/share/man  
lrwxr-xr-x  1 vpatryshev  wheel   10 Dec 10 19:10 /opt/share/man -> manuscript
```

This is kindergarten. And everything's like that there.

BTW, Eric still trying to argue with me. He writes that I wrote `git clone` incorrectly. What's interesting, he seems to always know what I did and that I did it wrong. He seems to ignore my comments. Does he have mental problems? No clue, I'm not a doctor.

Ok, who cares. I'll play with it tomorrow, and will try to fix it.

By the way, Eric already wrote to me that I'm reading an obsolete document (that's the document he sent me yesterday). The right document is the one which Bill wrote. And why exactly did Bill write a new one, instead of editing an existing one? He probably does not know how to read. Programmers, or, to be more precise, half-baked programmers, they don't read... they can't read. They write programs, they don't read them.

December 12, 2021

Just for fun, I'd say, I did open my work computer, fixed a script there, now it started working, but... in about 20 lines it started misbehaving again. I wrote to Arie, asking, where can I take all this, docs, etc.

Bill writes that he didn't write any docs.

Eric asks what permissions I have for the folder into which I store a link. Ok, I got it now about Eric. Will try to extract information from Bill. And it's time to start looking, what other cases do I have in Jira. And I closed my work computer: it's Sunday, after all.

December 13, 2021

Hardly made it to the meeting, at 12 noon. Nothing interesting, just the team's achievements etc; then it was my turn. I said that the API is already written, but I need to run integration tests, and nobody tells me how to do it. So it was decided that in the end of the meeting I'll share my screen and show my problems. Okay, and it arrived, the end of the meeting, and everybody got together to look at the talking dog, and I share the screen, and write `git status`; `git pull`; `mvn clean compile`. Here Eric screams: no, `clean compile` will take five minutes, and I have to go soon! Then Bill takes the word, and says: let me show you how I do it. I pass him the screen share (while it's compiling on my computer), and Bill turns on JBuilder, with letters of half-millimeter size, and I say to Bill, hey, I don't have a microscope (obviously Bill has a huge screen at home), and Bill answers: "you have to learn to read". And further on Bill, Sebastian and Eric went on discussing what's breaking in Production, and I was essentially forgotten.

So, what was the lesson. The team is pretty much disorganized, and I'll have to bug them individually, exclusively, get their knowledge, via Zoom,

or their advices (except "it works for me", it's not an advice). There is one advice, "launch postgres in docker". Ok, big deal, launching postgres in docker. And what next? Where do I get the database contents? Can anybody document it? I know the credentials, I know the name of the database, but where's the contents?

Another problem is that dependencies in pom-files are written with problems, and get resolved maybe every other attempt. This thing breaks, that thing breaks.

Okay. So far I can ignore them. Instead, I drove to the office, counting on finding this or that Chinese engineer, torture him, extorting company secrets. But no. Instead, there were, in the office, two Chinese girls, with whom we casually chatted about this and that and everything. You know, I adore Chinese girls. They are smart and strong. Steel inside, fluff outside.

In the evening I opened my work computer again, I planned to interrogate some people. But they are busy. Production broke. And what exactly is broken? It's Bill's code, `_.asInstanceOf[Map[_,_]]`. But see, I did send him a) a working code of that converter, and b) a test for that code, which, with Bill's version, crashes at exactly the same place, and doesn't crash on my version. So, I had publicly written to Bill: hey, here's my pull request that fixes the problem, and it contains a test verifying it, and I did send it to you last week, for a review.

In short, not my problem now, let them fix what they repaired. And I'm going to do my home chores. Namely, prepare to moving to my beloved South Carolina.

December 14, 2021

At night I suddenly figured out that I have several more ways to experiment on software. For instance, run tests on Dev servers. Or right within Jenkins. In short, necessity is the mother of invention.

But the problem now was that builds fail too now. This or that... they fail.

So I went to take a look, what's up there. And also, Bill had suddenly surrendered regarding his bugs, and he agreed to take my solution. But instead of approving my pull request, with my code and a test, he had copy-pasted it to his branch and merged it, having written on Slack: "took Juan's solution". Nice, nice. In the old days I would have been pissed off, but now I don't care. It's fine, let him get used to following me. And, what's important, he also took my test! That's a technical progress.

Meanwhile, I was experimenting: `mvn clean test` doesn't create jar files, and dependent modules are browsing all the repositories, while `mvn package` works, all the jars are created, but tests don't run

So I started bugging those two Chinese geniuses again. Like, here, how do we build it? And their answer was: but what do you want, run tests for all modules? It takes three hours. I write them: three hours okay, it suits me, let it run in one space while I work in another. And here they suddenly show me another command, `mvn install -T 4` What is it, "-T 4", they don't know themselves, and there's nothing in the docs. Actually, it's an experimental feature, "run it in 4 threads". Wow, it works, I like it. They also suggest to run tests inside IntelliJ. Well, I'm not a devops, to fix them the whole environment. But I can run the whole folder of tests, I don't mind. So, I'm fine with this. But oops, their code doesn't find Postgres - but I think I can fix it.

Meanwhile, I was reading the code, here and there; one Indian guy with a PhD (he doesn't work here now) wrote a dozen of huge regexes to parse documents. He has, specifically, a problem with parsing dates (no tests, of course).

I hope I'll fix it later.

Also, they are bugging me now that the functionalities that I implemented are not visible via openAPI. Now I write them, where exactly is this openAPI? The link in the docs sends you to a 404. And Sebastian writes that, hey, ok, the link in the docs is obsolete, we'll need to fix it. I answer: yes, it is very important to keep the documentation up-to-date. And now I see, who, he's fixed it. So I follow the link. And here I'm in a circle. They send me to a login page. I log in, and I'm redirected to the login page. And so on. I write them, and they answer: "because you need access". Now I'm fucking using my sarcasm, carefully. I write to them: "I'm very sorry that I made this mistake. Please advise me where I can get

access to openAPI (you probably appreciate the word "open" in this context). And Eric answers: "I've just gave you access". I bowed politely, on Slack (there is such a smiley there), and followed the link. Found my API, and sent the link to it to all whom it may concern (TWIMC, that is).

Now Ben writes me, asking me to make Business Unit optional in one of the requests. To me, it sounds pretty stupid, but we are in a transitional period, and the realms isolation is not fully implemented, so why not. I implemented it, and also fixed the internal language for predicates, just a little bit, while nobody's watching. To be able to write expressions like `predicateOne and variable("a").equalsTo(constant> B)`, this kind. It's pretty easy to write expressions like this, right?

Sent it for review to Bill and Ben. Although it's Ben who needs it. Bill approved it right away and writes me that I'm too polite, I could just merge it without an approval. Yes, I could. But I don't want to. (Actually, I did merge one such pull request in the morning, tired to wait for an approval.)

Okay, so I merged it, and Jenkins is telling me, post-factum, that the tests on master still fail, like they did before.

Oh fuck, I could have looked first. Since we have everything done manually. In short, I decided not to deploy my API, let them fix their tests first, or, to be more precise, not the tests, but the environment for testing. Wrote to Ben about all this, and that was it for today.

December 15, 2021

Well, I still could not find a way to run integration tests. But, on Ben's request, I've updated one small thing in my code.

Also, found that the command(`mvn install`), that they advised me for the "right build" does not check the style, so my merged pull request got an error (120 characters in a line of code), and our half-devops (they are, formally developers, but irl they are devops), with a vengeance, rolled it back, and David had proudly sent me a reproaching message about these 120 characters. In response, I wrote to him, saying, hey, David, can you show me what should I run before merge (actually, I believe it's them who should run it, before a pull request can be merged) - and, in response, a silence. Sure. Why should they know how to run tests or style checks.

Still some shit happening with the tests, and Ben is writing to me that he's still getting null in response to the requests that are, theoretically, sent to my service. No, interesting, whom should I ask?

And another urgent task was assigned to me. Seems like I'm supposed to add, in my stuff, links to four structure: that's what Jordan ways. But then it turns out that it's no tlinks, but I'll have to insert JSONs into some fields, four or fix, hard to tell. So I wrote to Jordan: hey Jordan, let me just add a map, call it "extra attributes", and you'll be storing there whatever you want, storing and extracting

too. Because, see, you suddenly need it urgently, and what if tomorrow you'll need something even more urgent, and a more sophisticated JSON?

So, you see, it's a circus. And I wrote to Krzysztof: hey Krzysztof, are you coming to our circus? We have clowns showing tricks. But Krzysztof is still choosing. He's a spoiled boy.

December 16, 2021

I'm tired a little bit.

The Chinese geniuses got angry that I added functionality to "their" code, without their approval, just with comments. They wrote a bunch of shit, like "I don't understand how it works"; I wrote to them: ask the questions, I'll explain it. They are pretty ignorant about shell scripting. Not that I care; I don't have to depend on them, and they on me. I told them, if you want to roll back this merge, go ahead, and I'm too busy now (this is a Chinese trick, "I'm busy").

After the meeting we stayed, with Stephen, and Bill, suddenly, showed us a bunch of tricks: how to run integration tests, and how to run end-to-end tests. I tried it, and it works. Wow, fine! Meanwhile, they wanted a couple of different things. The client sends a request for a list of something, which my service returns, but the response contains a null inside. I don't send anything like this. That's number one. The second issue is that our Frontend people have suddenly change the record structure, and I have to make updates.

I started with the issue number two: added variables, and made the rest to be verified using reflection. So I've only changed some constants. Cool. As to the issue #1, I did, as Bill taught me, send a request from my computer to the Dev server, and go this null, and looked into the server logs (Bill told me where the logs are, and I did `tail -f log | filter MyServiceType`, and I saw that I'm concatenating two non-existent filters. Ok, now I see, and I sent a pull request for fixing it, and finished it for today.

Oh, also, I had a 1-1 with Arie. It wasn't long; I asked him whether anybody complains about me. No, he said, I should bug them more. Oh, nice, that's what I'll do.

December 17, 2021

This day consists of meetings.

But first, I found that Bill had approved my pull request regarding list API, and I, why sitting in the meetings, merged it and deployed to Dev.

There was another pull request; Jordan had approved it, but during the meeting our Holy Prasanna expressed his desire to get for his personal approval every change in record structures. So I sent my pull request to him for review, and got a bunch of pretty good, meaningful suggestions (instead of

Jordan's hack in which some structured fields are represented as JSON texts). Wow, nice! I thanked him, and, happy wrote to Jordan, asking whether he's okay with this solution? Can we do it this way? By the end of the day Jordan wrote that we need to discuss it, and let's do it Monday, since he's busy now.

Meanwhile, Arie asked me to merge this stuff today. He has also planned that I fix the list API, but sorry, my pull request would not be enough. In the end I had promised him, vaguely, that I'll fix it during the weekend. It's impossible to do it now, because Jordan is doing something on Dev (sorry...)

At that "big meeting" Prasanna was saying remarkable things. That our build process is unstable and not well-documented. That we can't run integration tests. That because of all that shit, like, pushing fast to deployment, the technical quality of the code deteriorates. And why t.f. are methods names like almost 100 characters long. I was ready to applaud to each thesis. When did I hear any such thing, not from myself? Probably never. And we should not call a data type "BusinessUnitConfig" if the record as the data about Business Unit, not about some kind of config. Wow, wow. I thought we have a bedlam here. Not exactly. It's just some patients have problems.

Wow. We will see.

By the way, the meetings were happening while I was on the road. So that the connection, was often pretty flaky. And then my work computer was discharged. Actually, we may have agreed on everything... or am I hallucinating?

December 20, 2021

Oh, it was a work day today.

So, what happened? Prasanna had kicked the asses of our Frontend people who decide to design the API and the database. And then there was a meeting, at 10am, where a bunch of people got together to discuss what I should do, then they switched the topic, then again, and again, and the last half an hour they were busy planning the "sprint". Sprints here are short, Tuesday to Thursday. Monday we are planning, Friday we are reviewing the results.

Then we had a huge "all-hands meeting"; not that I remember what were they talking about.

Stephen wrote that he has a pull request, but so far he hasn't created a branch. Me and Bill, we spent some time trying to convince Stephen to first create a branch and push the code there, creating a pull request, and then show us the beauty of it.

The code is pretty much okay, well done. Bill approved it; I wrote a couple of optional comments. I like it this way, write comments not insisting on their implementation, but if I observe that things are not improving, I'll increase the pressure, and if everything's fine, everything's fine.

Found a bug in my code (while they are trying to agree on the record format). Jackson takes an object and takes out of it all that it thinks it needs. For instance, if there's a method `isValid` (which, in a good design, should not even exist, since invalid records should not exist), this means, for Jackson, that there is a boolean value `valid`, and it serializes it. It's no problem banning this trick, but it would not occur to me that things like this may ever happen.

December 21, 2021

What do we have today? Still Tuesday? It's bad... In short, yesterday we couldn't merge anything to master, because they were cherry-picking. But I found a bug there, and I needed to inject an annotation for Jackson, to make sure the method `isValid` wouldn't be perceived by Jackson as a variable called `valid`, and, since Eric understands the problem, he had approved it right away. Then I had to eventually do that stuff, "new data format". Our activists have eventually slapped together a document. I bugged them, why are structures duplicated there, why timestamps have a type of `String`>, and they changed it to `Long`, and why are they now `Long` instead of `Timestamp`, etc. I guess I got on their nerves.

Now David writes that `get`, `set`, `is` are not needed at all. What's your suggestion, I ask him. Eric suggest all kinds of stupid stuff, but he's a foreigner, a Chinese, he doesn't feel the language. Jordan also does not feel it, his English is a little bit broken; but David suggested just to kick out these prefixes. I didn't first appreciate his idea, but then I did, and I have kicked it out. I'll be short from now on.

But that's nothing; there's a panic in the team: Tuesday is the last day for some release, so everything's urgent. They wrote a document, now go ahead and write the code. At the same time you should participate at meetings. Ben is incessantly talking, something that I would not say I know what he means. Our smart Indian PMs are silent (one of them had confessed yesterday that he doesn't understand the table describing the data structure; I like this honest attitude.) Arie seems to be pretty nervous, maybe afraid that they would kick him out. Bill, when he manages to take the word, starts with his long sentences: after all, he's a SUNY professor, a former one (like me).

And all this is happening while I'm in the car; my wife is driving, I sit and turn on the mic only when I have to say something. It rains outside, I'm exhausted, and my sugar is constantly dropping, but why? What did I do?

They were very insistent, they want me to implement everything and attach WebSockets. WebSockets actually have nothing to do with all this; it's just a name; what we need to attach is a reporting, to send events on changes, and whether they go through WebSockets, that's secondary. There's a hierarchy of concepts, isn't there?

And they want me to implement everything by the end of the day. I started bargaining, and told them that I'll implement everything except WebSockets, so they assigned it to Sebastian. It turned out that

it's just a one-liner; no big deal if you know.

And what did I need? The idea is this: there comes a JSON from the client, with a crazy structure. I need to put it into the database, in the form of properties, actually, and to turn substructures to JSON. That's what they had invented yesterday. I don't see a ready solution, but the idea is generally ok. I know a better idea, but it seems improbable to sell these bullshitters an idea of flat properties, regular languages, relational data, and the like.

That's what I was thinking, and, meanwhile, they were discussing what else can be done in the remaining several hours. I made a promise (and I hold my promises... almost always; but I implemented it a little bit later tonight).

They also came to an agreement that we don't have time to deploy to Dev, we don't have time to deploy to Staging, and we don't have time not to run the tests, but even to write the tests. Let's throw it to Production. And now they are screaming, interrupting each other, and I listen to them and trying to figure out how to keep my professional honor without getting immediately fired.

At the same time my sugar is low, I can hardly think, and we are parked on a parking lot near Aldi, and my wife rolls a cart full of food, and she asks me to open the trunk, and all that like that. Oh, and we also had, but in the morning, troubles talking with US Bank, with Chicago Title, and so on.

Okay... the meeting is over. Bill started telling me how he had already solved a similar problem. His solution consists of adding String-typed fields, with JSON inside, but make sure they won't be able to go deep into JSON (for that they have a prefix in JSON, underscore). And when we fill these fields before writing to the database, and the fields from where we take the values, we set them to null. In short, it's year 1990, and the language is C++. Or even Fortran.

I was actually thinking about something like that, but without mutability, and have a mapping, lenses, etc.

But when we got to our apartment (that's in Myrtle Beach), I got a better idea. Have a separate serialization for the database. So. Big deal. I got the idea, I implemented it. And wrote the tests. It's 10pm now, and I sent them a pull request. It's 10pm here, and it's 7pm over there, in California. Let them study my code. It's a little bit complicated for them (Scala reflection, ClassTag); but it will be easier for them to just approve it.

We will see. Nice people, in principle, but with not much inside their heads. Well, they have a king for that, Prasanna.

Now I'll go have a cup of tea, will take a shower, and then check whether they approved my pull request or not. Or else what?

Aha, I had to write another pull request, because the one I wrote was already too distinct from the master, which was changed a lot today.

Doesn't matter, what's important is that nobody's angry. Even Ben, who needs everything ASAP, he, good-naturedly, quits the meeting first, and everything's just fine.

Everything's fine. I love sports programming. P.S. Ok, Prasanna took a look at my solution, didn't get anything, and suggested to discuss it tomorrow. Why doesn't he discuss the document, wasn't he going to put his signature on every database structure change? And? And now he likes the theory, but he's suspicious about the implementation. The idea that we pass around the data as JSON, not as ASN.1, seems to satisfy him. Probably. Maybe.

December 22, 2021

I'm still working, while it's almost 10pm.

Well, yesterday Prasanna wrote that he doesn't see my code on Dev. And that I sent him a pull request for a review, he doesn't see it either? In short, they had convince me, at the meeting, never to wait for a review, but instead just merge to master. Fine; why do we need branches then? We could just commit to master. No tests run on branches anyway, and neither they run on pull requests. Okay, now my hands are untied.

And I walk all day with my hands untied. There's even some progress. People, meanwhile, are casually changing the data structure - but I don't care! I have an abstract code, it accepts any structure. So there. Now, in the evening, I open my work computer, and see on slack that they are all on Zoom. Now I go to Zoom, and I see that they are editing my tests! Because you can't just change the data structure, you also have to change the test samples. Eventually they left it to me, fixing the tests that fail on their changes. Okay, I've fixed them.

So, in short, all that kind of small stuff, this and that. But at least they cooperate. Jordan cooperates, Bill cooperates. Ben slacks. Arie is watching, what else is not finished yet. It's okay. It's some kind of work.

At 2pm two Chinese geniuses set up a demo, "how to run integration tests". Except that they didn't show the trick per se, how to run them. They said "ask devops for credentials". But they showed a bunch of useful tricks, and I'll use them, as soon as I get distracted from deploying to Production. As soon as I fix the code, I'll start testing it. The way they do it in Quora, where they first deploy to production and then start testing.

During the breaks we walk along the beach. It's good here!



December 23, 2021

Well, the bullshit continues, but now I seem to be on the winning side.

Last night, till about 23:30, instead of investigating and fixing bugs, I was working on Prasanna's instructions, which class should be returned as a response, and the like. Theoretically, I had introduced union types, because a response may be an error or contain a data record, but Prasanna, I'm sure, never heard of union types, but he always Knows How Things Should Be. Not that I care, they pay me not for a show off, and not for the quality, but for the quantity. I'd love to work, with a great pleasure, somewhere where they appreciate the quality and don't get crazy, but where are such companies? There was one, but it's closed now. So, my honeymoon here has ended. And thank you so much, the honeymoon lasted for three months, this is unusual. Now I'm fine, no overworking (you wish...), no initiatives. I can't sell anything to anybody here; everybody thinks he's a genius (yes, "he", because female programmers don't exist in this company), and they don't get into discussions. Especially you can try to convince the Chinese geniuses of anything. Oh fuck, whatever. I have a book to finish editing, instead of trying hard to implement CRUD. Selling them software development culture... impossible, and I'm too lazy now.

In short, we had a meeting this morning, and Arie suddenly asks whether Prasanna approved this record format. And I answer: Prasanna stopped complaining that everything's wrong, and I, I guess, have fulfilled all his desires. Still, nobody has approved my pull request so far, and yesterday I was offered to merge it without an approval, so, it's still a bit of a mystery. What should Prasanna sign, the code or the document? And Arie responds: Prasanna is supposed to sign the merge into Production.

Then what, then everything's fine, all done as he wanted, except that the `List` function still returns bullshit I'm working on it. When will it be ready? - asks Arie. Today, - say I. - Are you sure about today? - Yes, I'm sure.

So there. Next they continued discussing, and I started investigating, how come it's still broken. Ben laughed at my sarcasm, and others didn't get it. That's how it works. Napoleon, Hitler, Alexander the Great, Jobs, and others, they lost for the same reason for which they were winning. This is typical. The same happens on the stock market. The hubris that leads to getting detached from the reality.

So, and Prasanna obviously thinks that he's surrounded by idiots, and only what he thinks is right; and those whom he doesn't understand, they are just confused. They invented some shit called monads, etc. We don't need no monads.

Meanwhile, I spent all days having sex with this amazing bullshit. For example, if you request "give me records of this type, here's the filter", it returns an array of empty records. That's because you didn't specify which fields you want. But as soon as you guess it, the rest is easy. But it's funny.

And besides, the names of the fields in the database and the names of the fields in web API are different. Why is so, who knows. I can ask Bill, but I'm not sure whether Bill has an answer. Prasanna

told him so. Bill wrote an automatic translator: he takes a field name from the API, splits it into separate words if it is in camelCase format; then he translates each word to “capitalized”, and connects them, with spaces as separator. So “camelCase” turns into “Camel Case”, and “instanceID” turns into “Instance I D”. This last thing, I have to fix it, as an ad-hoc.

And all this is like that. But at least I’ve learned to write unit tests for it. Added a bunch of logging into the service, so that I see in the logs what the database service returns; then I pull a chunk of that log to my computer, extract JSON from there, put it into a resource file, and kaboom, I can test ad libitum. Pretty simple, eh?

So that now (“to still the beating”) I’ve just deployed it, and everything’s just fine... except that, in the “new record format”, now we have all kinds of bullshit, like a python exception with a stacktrace, saying something like “a coroutine is tired of waiting”, etc: megabytes of garbage, hope it does not break HIPAA.

Now I notify Jordan, and he’s answering already: here, we have all this ugly API, and I, -says he, -wrote a new document. I informed him that I have “everything’s ready” (it’s funny, kind of, since there’s a bunch of things to do and even to implement, but they can already start playing with it, the Frontend people), and I will read his doc, and I may object, or not. And now let’s implement it. And he writes: let Prasanna approve it first, and we shouldn’t do anything before that. That’s smart. How do we know, does our phone need rounded corners or triangular ones, it’s only Prasanna that knows everything, and, by the will of Allah, he knows everything better than anybody else.

And the week has ended, thanks Allah.

No, actually, it’s not that bad. I made about a dozen merges to master: nobody stops me, and I can just add a couple of lines of logging and merge it. In an Advanced Bureaucracy this would take a day, and, in total, two weeks. But here, slam-bam, and it’s merged, and goes, if not to Production, then to Dev. Everything’s fine. And in general, since I’m inserting tons of tests, it gradually improves the culture. Not that it will lead anywhere, regarding the culture improvement. Well, I’ll talk to Arie, he has a philosophical approach, kind of.

December 27, 2021

A quiet day today. Which means that trying to get from our devops the right access to aws makes no sense today. I’ll wait for the new year.

So I started running another test case, and understood that something’s wrong there, but what is it? Hard to understand if you get a stacktrace in the form of a string. Well, what know, I wrote a tool to restore the exception (so that I could throw an exception and see what’s inside); wrote to David, our ideologist, he was happy about it, and I sent him a pull request. Why is it that I’m so slow? Now I wrote a new test for that, and, strangely, the exception that I throw manually, from a test case, contains very

unexpected lines in the stack trace. It's not clear where is it coming from (it comes from a runtime, but where from exactly?) I'm too lazy to deal with it, so I just wrote a function to compare exceptions more carefully.

Eventually I found what was failing there: an id was being duplicated. No big deal, I've fixed the test case, and everything runs now.

As to that id, Ben and Stephen started bugging me, how it disappeared in the record (it is now a method, not a field). I wrote to them: guys, let's update the document, and say that the record should have an id field (but why do we need it, if our id is synthetic?), and let Prasanna approve it. We do have ids, and you even get them for your request. But you can't change it.

We will see. What's good is that there's almost nobody there today; it's quiet, and there's a progress. I like it a lot, when nobody interrupts me.

December 28, 2021

WTF, it's already half past ten.

While we were driving from Myrtle Beach, I was still working.

Stephen called I thought it was something about "my stuff is all wrong", but no, it was a serious talk, we discussed some topic, forgot which one.

Then there was a meeting, and Ben started pushing the idea that all these entities have to have an id. In this company the whole architecture is that an id is a reference to data, and does not belong to the data per se. I started pushing this idea, and here we found a compromise. He says: I don't care what you have in the database and on the backend, but on the frontend we need these ids as a part of the data structure. Well, okay, agreed: I add ids into the record, and he's not giving me advices how to organize data. Everybody's happy.

Now I implemented all that, it was not elementary, because it was a little bit unexpected; and more, all my debugging is not local yet, but after deploying to Dev. But it's okay. Ben seems to be happy, I'm happy, everything's fine. Stephen is happy that we are now getting a good Scala code.

December 29, 2021

So, it's after 8pm, and I'm still working.

Shouldn't I move my work hours to an earlier time? See, there's no point opening the work computer before 11:30, but now, at 8:30, I'm the only one still working.

But before that, we had a meeting, and we discussed with Ben, what do do with the fields that disappear. I'll work on it.

It also turned out that Bill has slapped together so-called “field names transformers” (it’s a known stupid trick, we had the same kind at “We Sell”); and Bill’s approach is pretty creative. E.g. he takes “instanceID”, translates it into “Instance ID”, and then translates back, to “instancId”. He could, of course, check the isomorphism. Shit. And also, what bugs me is that programmers don’t know the difference between Id and ID (google it, google it). In short, taped it with a scotch tape, will fix his transformer later on. Because now, still, it’s not clear where do the substructures disappear. The unit tests work ok, but they don’t talk to the database, and the database is on Dev, so I have to run it on Dev, and so on. If they just gave me access to the Dev database, but no, I have to deploy. On the other hand, the positive side is that now I have tons of free time: I can unload the container (we have just moved), and attach a shelf, and clean the pool. While it’s building and deploying.

Okay, I’ll figure it out with the substructures. At least nobody prevents me from working, that’s good already.

December 30, 2021

I was having an idea that since substructures are not serialized well in the database, we can turn them all into flat files. But Ben had convinced me that it makes no sense, and using JSON is a regular practice. So I was doing it today.

By the end of the day I suddenly discovered that this thing, deserialization of substructures, wasn’t covered by tests, and it wasn’t working. Oops. Ok, fixed it, deployed it.

Meanwhile, on the server, my log records stop getting into the logs. And what am I supposed to do? Just for testing, added into the response class a field named `details`, and now I’m returning logs to myself. And to make sure I’m running the right version, added a test message type, where the server returns a 500, with a test message. All these tricks have helped me A LOT.

Of course, it would have been better to run the server locally, but devops are on holidays, and there’s nobody to bug, and these Chinese geniuses, they were not responding even in the normal days, why, to run the code locally I need to login to aws: they didn’t even understand the question, and repeated, “for idiots”, that to run a server locally you need to log in to aws. Sure I’ll fix it, but first I have to make it work, at least partially.

That’s it for today. It reminded me the old days, when, for debugging, when I was connecting some code in “protected” mode with some code in “real mode”, and was debugging it, the only way to communicate were pixels in the graphic memory. Like, a green dot, a red dot...

But it’s okay. I’ll finish it tomorrow. Although it’s a holiday.

Ben already told me that the UPDATE operation does not update anything. Okay... I don’t even have a test for that. It’s okay; tomorrow.

January 2022

January 03, 2022

Almost nobody at work today. Just Stephen and me.

In the morning I rushed to write a unit test and a mock for this service (partial mocking: one dumb bitch, Camie Hackson, at Google told me once that it's an unacceptable practice). I created all that, ran it, fixed the bug, sent it to Stephen; he liked it, approved it. It's probably an exception in this world that we encounter normal people. These two Chinese geniuses, it would take days to beg them; Bill is busy 24 hours a day, remodeling his new house.

But still, it does not solve the main problem: updates don't work, there's something wrong in the database service. Why is there a service for that? We could just have a library to access the database. But now, we run all those services, all around.

And if you saw that code. Reminds me year 2015.

Okay, whatever. It's just that there's no harmony (in this world).

Meanwhile, Stephen has slapped together good config files, now we can run integration tests! But I haven't touched them yet.

January 04, 2022

This morning I continued my investigation, where is this `invalid_request` coming from, if I have injected into the service a bunch of statements like

```
if (request.toString contains "TEST_...") throw new IllegalArgumentException(instanceto catch, where exactly this invalid_request happens. Couldn't find anything. Suddenly, foundthat it happens between two pretty innocent lines.
```

Now I went to check out the logs. Suddenly my log lines are showing up in the logs. And there, eventually, I found it: the database (to be more precise, a Man In The Middle) throws an error, and the message from that error ("unmodifiable field") is only visible in the logs. Oh shit. Ok, I'll fix it for now, but I'll have to create a test branch, to investigate this horror. Written by the two usual suspects, and some David, who already told me that details of an error should not be shown to a user, because it would break the regime of secrecy. OMFG.

But well, ok, things are clearer now.

We had a standup at 1pm, and Ben was giving a long (but meaningful) speech. Then it was Bill's turn. And he was like, slooowly... "I was doing something these days... will try to recollect what was it I was doing... I was doing something... but I don't remember what was it...". And he couldn't remember.

I planned, when it would come to my turn, to talk about deconstructing the service, so that now we can, like other services, test it with unit tests. But there was no time for me; Ben interfered again, and said that we need to slightly change the logic of updates, and that they'll be sending not the whole record, but updates for individual fields. And how, - asked I, will we delete the fields? Oh, - he said, - we won't do that. Well, that's fine then. (I'll write it the way that deletion would also work, and let them write a test first to catch this bug of possibility of deletion.)

In short, we spent an hour buzzing, and after that, immediately, the two Chinese geniuses told us that from now on we'll be making service mocks, and it will allow us to test services using unit tests. Here I turned on the microphone and said, cool, I had implemented it yesterday, and it is already merged to master (and I have an approval of that pull request), but, as Hoffman described vividly in "Klein Zaches, genannt Zinnober", nobody paid attention to what I said.

Later the same geniuses wrote to me that it was Prasanna who told them that updates should not include deletion of attributes, only changes (as if replacing an attribute with an empty string is not (virtually) the same as the deletion of its value). That's where all this is coming from. From the Wise Leader.

Kindergarten. Circus and clowns. If I cared, I'd be boiling already, talking loud and fast. But now - no, I don't care. Also, I've just watched "Don't Look Up", and it's brilliantly illustrated in this movie.

January 05, 2022

Strangely, I worked since 6:30 in the morning to 7:30 in the evening (and I still have to deploy something).

That's because Jordan, last night, had slapped together a new API format, much more reasonable than before, and he had documented it perfectly, and, as a result, I had to update my code. Good that have all this stuff covered with tests; all I need is to parameterize this and that. But still, I will need to write some code.

So that's how I sent a pull request at 7:30pm, and notified them on a general channel on Slack: Arie was already asking when it will be ready, and Ben was asking, and Jordan was asking. As soon as I sent it, our Prasanna, from high above, suddenly shared a very very good document, "how to design API". They expected from me that I'd implement an API today, but now we have a new party program, so there.

Actually, I like this kind of work, so far. In the morning somebody asks me, "when will it be ready", and I answer: "tonight". And nobody, so far, bugs me with "why this, why that". We need it, ok? Stephen mostly just applauds ("till thou applauds the deed"), since he understands Scala. Bill, on the other hand, writes a funny nonsense, like "I saw this scala trick in your code, this: `PiService` => - what is it? Or" I'm not using traits yet..." Is he too busy to read a book? Scala is almost 14 years old.

Oh, whatever. The most important is that they don't interfere. After SaveOnHealth, in every company they were interfering, and complained about my sarcasm.

January 06, 2022

So, today I tried to do my best to stop working before 12:30pm. That is, 9:30am on the West Coast. It kind of worked. I turned on my computer, started working on end-to-end tests, to test the new API - but a meeting started.

It turned out that /get in Ben's code doesn't work. I promised to investigate it. Arie asked me, when can we expect the results? Ok, today, most probably.

Jordan writes that one synthetic attribute is being wrongly synthesized (why do we need it, if we concatenate two fields with a space char? We need to use the space char, and I was doing it via underscore).

Ok, sorry, I fixed it by the end of the day (had also to update the tests for the new format). UI people were nagging me that I added, on Dev, about a dozen test records to the database. Well, I thought it's Dev, so what difference does it make to you? But eventually they had convinced me to create a new "project", etc. I did, but it was too late to configure it. Tomorrow.

That's how the day has passed. Stephen sent me a pull request. I sent him an advice to improve one small piece of code, to make it more Scala-idiomatic.

They were also discussing some pretty weird shit: Bill suggested to keep, in the field, three variants: Some(something), None, meaning the lack of value, and null, meaning the lack of the field. I suggested not to be too creative, and just have a parameterized type for this kind of circumstances. (And, generally, just ask me first, I'm a Scala guru here. But... Dunning and Krueger prevent them from asking an expert.)

That's it for today.

January 07, 2022

Nothing essential. Meetings all day.

At the "demo" meeting, by some miracle I managed to have a demo, how i deconstructed a service and hacked unit tests for it, as a result of the deconstruction. Those who got it, they appreciated it.

But mostly the discussion was about "what's not working here". All my stuff works. The usual questions were: "what's up with our websockets?". Well, that code was written by the two Chinese geniuses, so have to ask them. But probably I'll have to deal with it on Monday. Because it doesn't work actually, and, instead of tests, I see statements like "I swear it works for me; did you turn on the VPN?"

Okay, we'll ping it on Monday, with Ben.

January 10, 2022

It's Monday today; good, nobody interrupts, so I was just working. Wrote three and a half of pull requests. The last half is compiling now; tomorrow I'll merge it without review, it's just about code style in the tests.

Otherwise, I took a look, what do I have in Jira; most of it is done.

Only Stephen is visible on Slack, and he also approves my pull request real fast. Bill is sitting quietly, doing his stuff, not reacting to other people's pull requests, and only continues slapping together his nonsensical code. It's hard to work with him. Well, whatever. I'm splitting my stuff into pieces.

We had an "all-hands engineering meeting" at 2pm; everybody was reporting... no, just the managers. As to my class busv (it's an abbreviation), they asked about it separately. I did report, but, I said, it's not clear about the notifications, and besides, the only tests we have are unit tests written by me, and it's not clear, what's where.

Now Eric pops up, reporting: "I've fixed everything Friday! Fixed everything! (he has zero tests). They were passing around wrong ids, so the subscriber wasn't getting anything. The subscriber subscribes for an entityId, and it was wrong!" It was wrongly named in Eric's code, and I was passing him the right id. But Eric seems to be the kind of Engineer that believe that when a variable is renamed, its value changes. He's not alone. There are plenty of these in our profession.

Then I ask: "Eric, okay, suppose the id is wrong, but should the client bother what's the value of the id if he doesn't receive anything?

Now Prasanna interferes: "the client subscribes to certain ids, and that's why he doesn't receive anything". Now I ask: "Prasanna, how a client know the id of the entities that are in the process of being created, which id does the client provide, an imaginary one?"

Prasanna explains: "you don't understand, the client provides the id during subscription".

You know, when someone tells you that you don't understand, it means one thing: you are dealing with someone who doesn't bother to understand, and he projects.

In short, "let's talk after the meeting". Ben wasn't present. But at least I figured that during subscription one has to specify some filters. I have to investigate this subscription process. Have to talk to Ben. I don't think I can help him, though: Eric, being a Chinese Engineer, won't allow me to fix his design to make it meaningful. Let them do it. We have Bill. Although Bill is a white cisgender, he behaves like another important Chinese Engineer. No point to dispute. And why should I, I'm fine where I am.

And so on. But still, so far, enough freedom there. So I'll be gradually fixing it all.

January 11, 2022

In the morning I decided to check how the operation `delete` works. It doesn't. So I started fixing it; and also, found another bug (the server returns an error, because the fields are not fully serialized; so I wrote an implicit class, to make it understand what to do about it).

Also, added an attribute to the Bill's class, to handle stuff properly; this attribute comes from the client, so something should be done. And we should stop calling a three-component key a "pair". It's a key.

In the end I was asked to practically deploy to production. To deploy a cherry-pick with my precious updates. But it was them who were cherry-picking.

So I had to get familiar with this fucking technology: not to support the code as a whole, but in pieces. It turned out pretty primitive. Say, I have a commit, here is its hash, and what next? Next we take Production and insert there (cherry-pick) this commit. Obviously nothing compiles. Now what's next? Next I take the fresh versions of these files, from master branch, to make sure that the code is adequate (compiles). That's it. Of course it's not a cherry-pick as it is known in our profession, it's a profanation. But the problem is solved. 16 files get deployed to Production, via a pull request whose code review took just five minutes for these two Chinese geniuses. I like it.

Let me quote Jordan, from our meeting today. Jordan said: "My code is ready for testing; I wanted to test it, but the CEO told me to first deploy to Production".

Also, they discussed at the meeting, how notifications should look like in "websockets". I, rather impolitely, told them: I know how notifications should look like. Gave them a short lecture. After which the PM says: but we don't know yet whether we need notifications. Well, - I say, - I know how to do it, and you know whether they are needed. (And why do we have to discuss how to do it?)

Anyway, everything's fine. The job is good, I like it. Something like a vacation.

Actually, I'm working on page 122 of my book, about Goodstein sequence; I have to explain it so that Bill V. would understand it at once.

January 12, 2022

I just worked. Was deploying to production. Fixing bugs. Everything is so unimportant, and so slow... In the free time I was working on my book.

January 13, 2022

I know, you can't expect a thank-you. I sit and work like crazy, while Arie says: "a month ago we had decided that you fell asleep at a meeting". As if I hadn't told them then that I have a problem with my audio. I did.

And also, people complain about me. Jordan complains that I didn't understand his documentation. Sure I didn't. First I did what could be simpler, then I read the docs again, and did it the way he asked. But he didn't like it that I dispute with him. So what?! Also, Eric complained that I, either yesterday or today, was not online at their (pdt) 9am. That's really funny. I work like crazy, but at times I have to a) go do something at home, and b) shouldn't I have some time for working, not just sit on Slack and answer immediately to all kinds of nonsense: it's Slack, after all, not a telephone. And what is it all about? I'm a little bit puzzled.

We'll see. Krzysztof decided not to apply to this company. Maybe it makes sense.

But wait, should I also start complaining about someone? My culture is different. And I live in South Carolina, where people have more decent attitudes. I'm not going to complain about anybody; let them kick me out, I'll go to the next company.

Currently I'm sitting, running the build (the end-to-end tests did pass, everything's ok, except that I didn't like the server message, it's not well-formulated. Now I'll slap together a cherry-pick pull request, and will deploy it to Staging. I shouldn't work too much. Nobody will thank me; I work for fun, but there's more fun when they appreciate it.

Fuck them all. But... maybe I should start interviewing. I see recruiters sending me pretty sweet positions.

January 14, 2022

Not feeling well. First, I'm sick. Second, I have to work.

Jordan complains that he send a POST via curl, and he gets garbage back.

I try to reproduce it, adding a test case, and my test case behaves! returning as expected. Now what? Now I started adding good logging to the service, how else can I do anything? And, after writing to the database, I read back from the database, to check.

By the way, one of such tests today, *for a while*, when I was creating one record (with a unique primary key) was telling me that this is a duplicate. Then it stopped reporting a duplicate.

Arie believes now it's a concurrency issue. Nice, really nice. But why does my test pass, and Jordan's curl doesn't? I also have to slap together this huge curl command: it's been a long time since I wrote such long commands, with tons of headers.

And it's Friday, five hours of meetings. And at every meeting they asked me when it will be ready. And I'm not feeling well, because yesterday I got a covid shot.

In short, no fun. Just no fun. It's 8pm now. And I can't work, will work on weekend.

Meanwhile, I was too busy with Goodstein sequences, and started (while it builds/deploys on the work machine) writing the code for this weird Goodstein. Oops...

January 15, 2022

In short, it's some kind of masochism (hi, Leopold!). Yesterday I wasn't feeling well, but continued "doing my job", and continued it today. The idea is "but I promised!"

The but was this: A client side sends to the server some kind of JSON. In which, if a field of a record should be deleted, it says "thisfield": null; and those fields that we don't change, are not listed.

The smart piece of shit that converts JSON into an object with a specified type (which is published via openAPI), if a field is of type Option [T], and it's missing in JSON, it becomes null; if the field is null, it becomes None, and otherwise it becomes Option (something).

I think you'll agree that this feature is totally dumb, and is not documented. Bill wrote something like a "test" (meaning, it calls a function and prints a result: Bill doesn't know how to write tests); and bill is aware of that, but he never told anybody.

And so on.

And the funny part is that when I send my request, it reaches the server and is properly processed (it gets through as JSON); and what Jordan sends, is also JSON, but he's getting garbage in response.

And so it goes.

It's over 9pm now, and I sent a cherry-pick pull request to Eric; he can read it Tuesday or whatever, this is not my problem. In a sense.

But fuck, you live, you work, you count on a common sense, and now what. Java people writing in Scala. They have some screws loose, and am I supposed to tighten all these screws, really?

January 16, 2022

It's kind of stupid, but I did open my work computer.

Eric had approved yesterday my cherry-pick for Staging, having written a good suggestion for a further improvement; yes, I'll improve it in a separate pull request, thanks, Eric.

Now I deployed it to Staging, and wrote to Prasanna, who was asking about it Friday evening, is it ready, is it? I wrote to him that it's ready. And wrote to Jordan, saying, hey, you can test it now. I can't test it, I don't have enough access to Staging. Well... I can ssh there, and the rest can be done right in Staging, via curl. No, it won't work. I will need a key to the project. Well, fuck it. It works for me. Like that foot, I too have a foot, and it doesn't work.

Tomorrow I'm not going to open my computer! Glory to MLK!

January 19, 2022

I got a feeling that, as the deadline has passed, everybody's quiet now. At today's second meeting it was found that there is some shit assigned to me, but when I asked whether I should put it into Jira, as a case, Arie said that right now no need to do it, and Prasanna wrote, into the doc, that it's low priority. Just fine! I was sitting there, hacking some shit, and listening to their speeches.

Also, my VPN stopped working. I asked the devops, on which channel should I ask for help? And they were so aggressive, like, why didn't I provide them with all the details? No big deal, can do. I restarted tunnelblick, sent them the logs. It's not my area of expertise, sorry.

That was it for today. Will do something else tomorrow. Oh, now, two of my pull requests were approved, and I merged them. I'm cleaning their stables. The thing is that their "core team" people are more or less happy from my actions, and our berry but boring guys, especially Bill, have a principle "don't touch my code" - should I bother then? I should rather stay as far as possible from his code. His code reminds me "*À la recherche du temps perdus*" by its volume and by the number of code repetitions.

Never mind, I'm now busy programming Goodstein sequences. The complexity calculations are even buzzing in my head. It's all degrees of 2. A degree of a degree of a degree... Even if you start with number 5, in about thirty billion steps it will be this kind of size, about thirty billion digits. That's just when you first delete the member of the first order. Okay, later, later.

January 20, 2022

I wouldn't say I understand what's going on. Seems like I have nothing to do, so all I work on is what I find on the back burner.

Namely, trying to run the servers locally. They say it's all documented. But it doesn't work for me.

As to the VPN, our devops were trying to get rid of me, until Grahame decided to help me; and it turned out that one has to log in to a portal, and there I see: my VPN password has expired. Could some script have sent me an email or some other kind of notification? Sad...

Otherwise, not bad. Hardly did anything. Sleeping, after a rush. I don't get involved in the discussions on how to organize data structures: I know that nobody will get my arguments (and I - theirs).

That's how the day has passed.

January 21, 2022

Five hours of meetings, and I'm totally exhausted.

So I didn't do anything, just sat in the meetings. One meeting started with Ben talking about how our Ministry of Justice blamed Belarus of state-organized piracy. Others were not aware.

That's it for today.

January 24, 2022

I think I started getting it, what's going on. I mean, the meetings. Or not.

See what was going on. Around 12 noon our time I opened my work computer, cleaned the Jira cases (nagging Bill regarding how come he does not “review” and does not close these cases), then I found a new, seemingly important, case. I refactored it fast, to add this small functionality (generating filters for the database), ran the tests, created a pull request, sent it for a review.

And now a scrum meeting. It lasts, as usual, an hour instead of a half an hour, and right after that we have the “big team meeting”, where Prasanna looks through the git merges on master and asks, “what is that”, and also urgent philosophical questions are urgently discussed. For example, “let's kick out data filters from the UI, or otherwise we don't have time (to work).” Also they decided to kick out my newly-added filter. I interrupt them, saying, guys, first, it's not a UI (our Indian PMs don't exactly understand the difference), second, I've already implemented it. They can't understand, how is it that a Jira case filed Friday evening is already implemented by 10am Monday (here, I'm afraid, I have just disclosed my capabilities, and it was a mistake). Arie writes: “Juan, what, did you already deploy it to Dev?!” I answer: “no, I'm waiting for the code review”.

But I was tired waiting, so by the evening I skipped all this “process” (which is not obligatory), and merged it. And I deployed it. And I'll just write to Arie, saying, yes, I deployed it, why do you ask?

So my dream seems to be fulfilled: the freedom of refactoring raises the productivity speed, and the high productivity speed gives tons of free leisure time.

But now I understand what's up with the meetings. Prasanna, the CEO, is bored, having nothing to do, so he gathers the whole company and interrogates everyone who committed something, “what is this pull request”. Some CEOs have this urge to interrogate. Look at Musk. But Musk has other companies, so he does not bug Twitter programmers every day, it was just a moment impulse. But Prasanna has nothing else to do.

January 25, 2022

This morning I didn't work much: we went to Lowe's to buy shelves for our home.

But then, before the meeting, I took a look at what they say on Slack. Oh, oops, the meeting is moved to 2 hours later, since Arie went to see a doctor.

I asked yesterday about `awsgooglelogin`, and they answered that it's not needed (these were the same people that have just updated the doc which says that we need `awsgooglelogin`). Ok, nice, funny. In the evening I tried the simple `aws login`, and everything works like a clock. My task here is not to achieve something by constantly bugging them, but to document the results of bugging and interrogating them.

There was a 2pm meeting, and Prasanna was quietly present, until he popped up and started screaming, how come server changes cause bad openAPI documentation?! - and I tried to explain that our test must test the documentation that our process generates - and here everyone got quiet, and we moved on to other questions.

Then they asked me again, how is my recent pull request doing. I explained again that it was yesterday deployed to Dev. There were no questions to me, and no requests. That shit that I had volunteered this morning, well, why me? they threw (reassigned) it to Bill. Probably Bill is jealous regarding features. Bill is a special kind of a programmer, but his smooth speeches are impressive, and the management loves his speeches, so why should I bother. I just sit there quietly: why would I care? You want some code, I'll hack some code. If you don't want any code, I'll work on my book. It was Kurt and Kinnar that taught me this trick: don't look up.

But no, I can't stop. I went ahead and fixed Scala coding style in a bunch of tests, and talked with David on this; David also sees that tests here is when someone writes an example, then he prints the results (`println`), and then he looks carefully at the `stdout`. The idea of automation rarely visits the heads of even the most Distinguished Programmers. They want to be in control, and automation for them is a pop art for the simple lowly programming masses.

FIIC. I can show them my scripts (like, how to deploy to Staging in one single click, this kind of stuff), but as soon as I tell them, there will be a critique that it's all wrong, and everything must be done differently. Same shit with AWS, these programmers have complicated relations with AWS, and they prefer complicated manipulations. None of them read "Amazon Pillars". Hardly anybody even heard of this document.

January 26, 2022

David is still dragging his feet with the review, saying "we need to run the tests". Well, fuck, of course. Yes, I do run all the tests. Because in our environment a created pull request doesn't run tests. They have an excuse: tests take too much time. Nice. BTW, not all the tests do pass actually.

Now I opened Jira, took a look, what do we have there. Closed a couple of cases as irrelevant, started working on one.

It must take about half an hour, but from the point of view of my coworkers it's a three-day job. We have a Thursday tomorrow; Thursday and Friday are fast days (not in the sense that we work fast, it's

in the sense that we are fasting, no merges in master.) So okay, I'll work on the tools.

My pull request was eventually approved. Then Jordan found a bug; I thought it was mine, but no, it was Bill's. Jordan threw it to me, I fixed it. But I didn't write a test for it. It's Bill's code, why should I write tests for him, am I his assistant? No way. I'm an important person too.

By the way, at times the company shows glimmers of consciousness. Suddenly David writes: "Juan recently wrote that we can't have SQL injection: let's fix it all!" - But the important Chinese engineers ask in response: oh no, why would we need it at all? David sends them a link to Wikipedia (I'd have sent them a link to xkcd 327).

Well... on the other hand, why bother, I have a lot of free time. I'm sitting editing my book; besides, I've built a shelf for my wife. And we had walked 14k steps today. Why worry? Be happy.

January 27, 2022

David has at last agreed that everything's fine now (I have all the tests pass, except those three end-to-end ones that the Chinese "team" wrote, and which fail on master), and I have merged my stupid fixes.

Started writing, because I had nothing to do, a converter, "class fields<->table of properties". Eventually I plan to make a simple converter for that, with no annoying details.

Also I tried to "follow the instructions", which were slapped together by various people, and recommended to me by Stephen, but it turned out it's all garbage, and Stephen sent me a link to his own instructions. That's what I'm going to read.

Also, it turned out that Magnus, the Swedish guy who was training the machine, at "Just Water", to tell a cat from a mouse (it was telling by the color, so it could not tell a gray cat from a gray mouse, and a white cat from a white mouse), now works as the same type of Machine Learning Scientist here. He's a nice guy, yes.

At 1pm we had another meeting, and a bunch of bs was being discussed there I was just sitting and listening (all my cases are closed), until I heard that we will have to be able to rename Business Units... then I jumped up and said, wait, wait, you were going to use business units names in composite keys, so, how can you handle this? Now Prasanna, who, not having anything else to do, likes to attend our meeting and give speeches, says: "well, ok, we'll think about it later", and I could not stop myself and suggested: "we need to have abstract ids, not using modifiable data as keys". And so on. It's an elementary. knowledge in data organization. but nobody here seems to have read any books, like Date, like Codd etc. Funny of course. But... should I care?

Then Ben wrote that such and such API is returning a weird error for a pretty normal request. It's actually Bill's code, but I already fixed it once, so I wrote to Ben that I'll work on it right away, but can

he create a Jira case for that, please?

But I had to finish my “converter” stuff; then there was a 1-1 with Arie, where he was mostly complaining about Bill, not naming him, but it was pretty obvious. He asked me to help Stephen to fix after Bill. Then he gave a long speech regarding our approximate future plans, what are we going to do next. My feeling is that all this activity, for the next couple of weeks, can be expressed in one SQL statement, called from the server. But I did not want to upset him, so I said: “we’ll work on it.”

Later I asked Stephen whether he needs any help. But he had just finished everything, and merged it, and deployed to Dev.

Having done with my pull request, I too went to Dev, to see how Ben’s request is failing (now I have a script for that) - but oops, Dev hangs. Dev hangs. I have another script, to ask Dev its version. It hangs.

So I wrote to the devops, describing the situation, wrote to Stephen, asking whether he’s aware that Dev is dead, and closed my work computer.

January 28, 2022

First, it turned out that I’m not reaching the servers because, out of three VPNs (!) one didn’t connect. Okay, shit happens. Stephen (the Vietnamese guy) showed it to me, and although they were telling me that two VPNs is enough, it turned out that no, not enough.

I tried, actually, right in the unit test that counter-example that Ben showed. Well, his request is wrong. It’s just that our response is not very polite, it just says “invalid request”. It could tell where’s the problem. Okay now.

Also I have inserted lots of tricks to the tune of “let’s replace with `classOf[T]` with `implicit tag: ClassTag[T]`”, and sent the pull request to Eric (all my tests seem to have passed); Eric approved it, I merged it, and started the build, and everything went fine, but later Eric rolled it back, because I had some errors regarding types. Well, right, but sorry, how come none of this is caught during testing or even compilation? Something’s wrong here. I’ll investigate this issue.

Then we had meetings, one after another. I was staying quiet, and only when Prasanna said that it’s time to start, before the deployment, to test all requests in the tests, you don’t need a server for that: here I interrupted and said that sure, but the problem we have not in the lack of the environment, but in the lack of tests.

But Prasanna is somehow too angry today. Your team, he said, works just at 10% level. Well, sure, I know. The biggest problem, I believe, is with his buddy Bill. Who complained yesterday that he had to work in the evening! (Fixing his own bug.)

I wrote to Arie that if I can do anything to improve anything, I’ll be happy to do it. He wrote to me that we need to discuss with Eric the idea of setting up classes for requests and for entities. I thought it

was about Swagger, but now. Eric wants to give priority to Scala classes, and the rest to be generated via openAPI. Well, okay, this looks reasonable.

So we spent the whole Friday at meetings (like we always do), but I also was slapping together some code, just to keep myself busy. Okay, I'll have to work on the problem that Eric found. It's unacceptable, the build process has to catch itself this kind of bugs.

That's it for today.

January 31, 2022

At times I have an unusual feeling that they are listening to what I say. I'm not used to it, and I don't talk much.

We discussed the problem of making a transformer from properties to an object of a given type. And I said: I've just merged a pull request where the `a method(Map[String, Any]).as[MySpecialRecordType]` does exactly that. I spent this weekend trying to figure out how can I use `ClassTag` for this. And here we go.

Sent them a link to this code that I wrote, so they could take a look how nicely it can be done. But I didn't start using it in the application yet. Let the dust settle.

Arie gave me a bunch of Jira cases, one (first priority) was especially outstanding: throw out the Python code that updates the database according to what it finds in the Scala code, and rewrite it all in Scala. I spent half a day perplexedly lookin into that Python code, trying to figure out how to do it without breaking anything, and wrote to Arie that, I think, first we need to document it properly. So I spent the rest of the day reading this Python. It's an interesting problem. Database diachronism, with no idiocy. E.g., how do we delete or rename something. Philosophically, it's kind of clear. But Python fails to handle it; everything breaks as soon as you start updating. Okey-dokey, we'll fix it; and it's interesting.

That's how I spent my day today.

Still, I'm curious, how come they are interested in my opinion? At the meetings, our CEO regularly asks: "Juan, what do you think?" Last time anybody asked about my opinion, it was in the USSR. Not that I suffered all these years: thanks to Kurt and Kinnar, I know how to enjoy a free time at work, the time when I have nothing to do. It was hard to learn, but now I know.

February 2022

February 01, 2022

Okay, I have slapped together a design doc, in the morning, and sent it to Arie, asking his opinion, and telling him that I plan to show it to Eric, Jordan and Bill. Arie responds: show it to Eric, he's familiar with the problem. Seems like, while Jordan and Bill are also working on this, they seem to be too practical, and too losers to be entitled to an opinion, from Arie's point of view. Interesting. Okay.

At 1pm, as usual we had a Scrum. Then it turns out that last night, at 9pm, there was a big alarm, because somebody deployed a copy of Dev to Production, and the APIs are different, and in short nobody knows what's going on, and since I was fixing one of these APIs lately, I was also invited. Jordan spent a night trying to fix it and figure out. I don't know. Jordan is a nice guy, but he's a little bit slow, and he's a Frontend guy, and explaining to him how to debug software is just impossible, what's possible is more panic.

Well, maybe not bad, but he complained to Arie that I didn't understand his doc (it's better to complain to yourself about yourself that nobody understands your writings). As my wife says, you are giving likes and good grades to everyone. Well... she may be right. I kind of sympathize with them all.

Ok, so, since now nobody is there to read my design doc, I started to update the wiki pages at work (nobody reads it too, it's probably just for me), and, in parallel, fixing stata style in the tests (simultaneously investigating what kind of shit we have there.) Sent one fix to David, and David is shar, and writes: "aha, you missed a space character, and here you inserted an extra space! And why didn't you rename this class?" What class, II was just fixing compiler's style warnings, I sent him 27 files, and there are 50 files left. In short, since you started it, Sisyphus...

Reviewed Stephen's pull request, and added a couple of more insistent advises to removed code duplication and use an existing function, instead of writing long lines of code every time. But I approved it. Stephen learns fast, and in general he's pretty advanced, so it makes sense to discuss.

As to eliminating remakes in general, it would be interesting to rewrite War and Peace in a compact functional programming style. Eh bien, mon prince... (Is Anna Pavlovna from Canada, with her "eh"?)

Krzysztof, by the way, has decided not to join this company, because if he leaves "Video2", they'll try to charge him for his \$20k hiring bonus; so he's going to hang there for the next five months. There's a problem working there, everything is under control of their Chinese division, and the lazy white people are not good at all, obviously

February 02, 2022

More news from the circus.

I was going to do something, and I even did something. But first, there's a new fact: in response to my design doc Eric answered reasonably that many things of that doc have been implemented, and we can proceed. He mentioned which classes are already there. Fine!

But then, at 1pm our time there was a meeting, and we had long talks about... wait not exactly. First, Bill was explaining, urbi et orbi, where these differences in interpretations are coming from. First, we have scala classes. We deploy them. (Because Prasanna told us to urgently add or remove some attributes). Then a special script generates an openAPI, and this information regarding the data structure becomes available on the internal web. Then another script generates (from openAPI) the Python structures.

That's when, during this process, an impedance happens.

I turned on the mic and asked: hey Bill, where are these Python structures generated, I have not seen this code. Actually, I was going to ask, but in the middle of my question Ben interrupts me, saying that we don't have time for empty discussions and get distracted, let's move to other topics.

Fuck. I wrote to Ben, on Slack, saying that hey, Ben, you yourself talk so much at the meetings, and you didn't even let me ask an important question? Yes, I wrote it, but we were on the road, in Virginia, and the connection broke, Slack turned off, and, in short, as it happens to me often, all my swear words evaporated in space. But I did back up my text, so I'll write to him. One can't behave like that with coworkers.

Then there was a usual buzz, then it was my turn. I have two cases "at the top of Jira", one is to rebuild the deployment system, so that the database onboarding be written in Scala, and it could extract the code from github or from the class files, doesn't matter. The other one is to add four new fields to BusinessUnit structure, it's very important.

And the PM and the manager are asking me, can I do it today? I tell them: the first task will take two or three weeks (that's where I'm a funny optimist, hoping that I'll have two weeks, not distracting me to shit like "add a boolean flag to a structure"). And Sridhar says: What?! Two weeks? We need it by the end of the week.

And someone asks: why do you need it? And Sridhar: really why do we need it? And gradually they figured out that it was assigned to me, for this sprint, by mistake.

Ok now, do I mind? And what exactly should I be working on? This one, add some shit to the structure, and, most important, "data ready for export". This flag will be set by users, from Frontend. And python scripts will be clearing it. Now I ask: there are many of these Frontends and Python scripts, won't they get intermixed? Are not we supposed to solve concurrency issues? Let me spend an hour or two thinking of it, and I'll design it how we can deal with this flag. (Actually, it's not a flag, it's a semaphore, not sure if anybody heard this ancient word or the name of Dijkstra.)

Then, after the Scrum, we had a "parking lot meeting", where I got Bill's answer, and were we started

discussing this flag, which is semaphore. I said again that I need a couple of hours to design a solution and write a doc. Because PMs and Frontend people don't solve such problems, and they should not.

Then the meeting had ended, and I started thinking. Actually no semaphore is needed here, we need a queue. Our designers, it turns out, never heard about queues. And while we don't have queues, we could probably introduce a couple of fields, updating them *in transactions*, and the "flag" will be a derivative. Now I'm ready to write a doc, and send it to Prasanna, let him approve or criticize, he likes it.

Now I wrote it to Arie. And he answers: implement it the way it's written in the Jira case. That's one thing. And another, I didn't stay after the meeting, although he was asking me. What? I did not see anything. And the meeting had ended. Well, anyway, the meeting is over.

As to the implementation, should I care? You've been warned, it won't work. I understand, you can't sell anything to Prasanna, and Arie won't support it. Fuck if I care. You need it urgently? No problem. And within half an hour, while sitting in the hotel, I slapped together the code and sent a pull request to Bill, for a review.

Now Bill starts asking questions: why this field? why that field? Why is it duplicated?

Later in the evening I wrote to Bill: here's the Jira case, it's all there, too late to change. Let the Frontend set the flags, clear the flags, the way they like. And we'll be logging these events: "flag set", "flag cleared". What's the point of arguing with idiots, right?

Seriously, it's time to move on. Where? Where people are smarter. What am I doing here? The salary is low. It's just that actually not much to do. And actually, I should not try hard. I have other things to do, and I have something to live on.

We will see. One recruiter is sending me pretty sweet choices. We will see. And I won't go invite Krzysztof, he'll hate me.

February 03, 2022

It's rather early now, 5:27pm, but I'm tired.

I sent a mail to Ben, asking him not to do it next time, and he kind of downplayed the issue. Actually, I don't care, I'm pretty insensitive, and he will probably remember. Because next time I'll complain to Arie. Whatever, let's forget it. We have a total mess at our meeting, nobody's moderating it, except for Ben, who interferes and starts talking and shuts up everyone else, when he's tired of listening. Is Arie present here, is not he? And besides, why does a half-hour meeting drags for an hour and a half. (Not a problem for me, since I have something to read, on another computer.)

Okay, and today, at the scrum meeting, they asked me again about the tool I'm developing, when it will be ready, and when I create a two-week schedule of implementing it (these people are clueless about

programming, they think it consists of blocks to be put together), I said that this case is not related to our urgent epic, and it was included by mistake; and now they switched to this one, where I added new fields, according to the management request: including that famous bit, “data ready / data not ready”, which I was going to implement the right way, but I was told “don’t touch anything”, because it’s the Beloved Leader Prasanna had dictated to the managers, they wrote it down in a crooked handwriting, and then copied into Jira, as much as they thought they understood.

But it’s okay, and I say, oka, I’ve updated this structure, but where’s the Jira case regarding the new request format? We can’t update new fields from the old request which doesn’t have these fields. And they said: it’s no time now to get into details, and Sridhar, he’s our PM, suggested to discuss it on Slack. I could have asked: will you create a Jira case, or do you want me to do it? But wait, if I start writing cases, I’ll be taking the job away from the PMs. So, go ahead.

As a result, I decided not to write a Jira case, and I just slapped together a primitive test case (with hardly any checks), and a fixture for that test, containing a sample JSON data. Now Arie says: before implementing something (that something is actually in code review already, and he even reviewed it), we need to write a documentation and show it to Prasanna. I rushed to answer: let me make a test to accept this sample - but then I changed my mind, and decided that I should rather update the docs and put that sample in that doc. Writing docs is no big deal for me, it’s faster than writing code.

So, I wrote that doc, copy-pasting a sample JSON from my test into that doc, and sent a link to all our wise leaders, and, it seems like, this was the end of my actions today. Sridhar is, of course too busy, he doesn’t have time to work. And I’m not supposed to do something before I get an order. Of course I have this mental problem. If I had enough freedom, I would have implemented it pretty fast and pretty reasonably. But while there’s no demand, there’s no response. I’ll be like that Soviet cosmonaut that did not see Good, while in space, because he wasn’t ordered to see God.

It’s time to move on, actuall. I’m exhausted. So I wrote to Abigail, an exquisite recruiter, who is looking for ZIO people, with a good money. I hope it’s not about Amazon, I don’t want even to try this highly-paid prison. Hmm, the day wasn’t over yet. When I got home, I looked into my work computer, and there was an uproar. Prasanna had thrown a bunch of ideas, how I’m supposed to change everything in that class. Bill, meanwhile, threw in a bunch of updates into that class, and, it seems like, he’s going to merge it right now. And Arie with Sridhar, they have discussed all the issues, and they have reviewed the whole epic, by which I was working, step by step. But these guys at least now wrote something meaningful and readable. And everything’s fine. It’s just a fog of war. Or?

Not that I complain. I find it funny.

February 04, 2022

“I’m still working on code changes. I still have two things that don’t compile”.(a quote from a meeting)

Ok, as Prasanna gave a bunch of Advices (stuff all this shit that I had added to the class `BusinessUnit` into a separate substructure), I, as a fan of playing by the rules, first updated the design dock, sent a link to those involved... but not time for writing code, we had meetings.

First, a “demo” where the Frontend people were showing their nice tables and expressed their sadness that there is no way to show structured data, like trees, etc. I was on the verge of telling them that they could have compound keys, with dots or slashes, and everything would be fine, but I know that making this kind of suggestions is useless; even if they consent to listen (which is hardly probable), they won’t understand anything.

So I was standing in that meeting, having lunch (we don’t have a table in our new house yet, so I was standing at the bar counter, and I was listening to them. Their design seems to be intentionally made so that you can’t read anything, brown letters on beige background. But it’s beautiful.

The demo was supposed to last for half an hour, but it lasted for an hour. In principle, we, Backend people, should also show our demos - but they wouldn’t understand it.

(Quoting ChatGPT:

Two mathematicians are walking down the street. One says, “Let’s stop and get a drink.”

They go into a bar. The bartender asks, “What’ll it be?”

The first mathematician says, “I’ll have a beer.” The second says, “I’ll have a beer too.”

They sit down. After a moment, the first one says, “You know, we really should have ordered differently.”

The second nods and says, “Yes... but they wouldn’t understand us anyway.”

)

At 1pm a scrum meeting started. As usual, it lasted one hour, not thirty minutes. They asked me how I’m doing and I said “I’m fine, thank you”. And then the Frontend people started again talking their usual nonsense and showing their brown tables (Babylonian style?)

The meeting actually didn’t finish at 2pm; it was just me, I closed my work computer, turned on mine, and went to Zymposium to listen to professionals talking about this-and-that-and-free-monads-typed-lambdas-Church-encoding. Keith Langdon is a great presenter, it’s okay that he had some errors, they are inessential. The ideas are great, how he from lambdas and Church encoding derives free monads.

At 3pm we had, at work, an “all-hands meeting”. I was ignoring them for half an hour, and went there only when Keith finished. At this meeting, as usual, Prasanna gives his standard talk about how we are doing exceptionally well, and how everyone should be doing everything right.

Then we had a “scrum planning” meeting. We have weekly scrums. They asked me again whether I’m already working on this tool for database onboarding for the new code. Seems like they have already accustomed to the fact that at times some things take me not half a day, but two weeks. They have not used, though, to the idea that they gave me a bunch of urgent stuff to do, and so I did not do anything for that tool. But they somehow believe my deadlines. Yes, during these four months I was following deadlines scrupulously (I just love this work style). So now they are giving it to Bill. Bill will finish it within two evenings, and then spend half a year fixing it, to make it compile. Bill is an optimist, and he always has his excuses.

Instead, I got some cases from Bill: “insert this here, insert that there”, etc. But... as Sri says... (who’s this Sri?), that first we have to finish the previous epic. Do I mind? sure, let’s do it.

By the evening I sent them again a pull request, with the updates according to Prasanna’s ideas.

And there’s Eric, an important Chinese guy, he doesn’t trust Bill for some reason, he doesn’t believe that Bill will do it, so Eric set up a meeting today regarding that tool.

But it’s Friday evening, for God’s sake. What meetings? We went for a walk, met Taylor, talked with him. The weather is nice and warm. So I suggested to have a meeting on Monday, like what sane people do.

It’s about how someone, at a meeting, suggested a bright idea that this or that feature is required by Monday. And Jordan, from Denver, says: no, I don’t like working on weekends, I tried it recently, and I didn’t enjoy it.

I think we could have done twice more if we had twice less meetings.

But who cares. If I fine one more job, I can work there during all these meetings.

February 07, 2022

When they gave me this new case, “create a generator to synchronize the database with the code, and also with openAPI”, then I asked Eric what does he have on that, and Eric told me that there is such and such class. After which Eric dropped everything and started writing all this stuff, helping me (while I was busy with other things) and Bill (who also decided to work on this issue). Now there are three of us.

Today I wanted to check how my pull request is doing. Eric had inserted a bunch of his annotations, into the class that we both were editing with Bill, in the same PR (editing the same code), so now I can’t merge anything, after his changes, and I’ll have to write a new pull request.

This is the habit in this company: as soon as someone starts working on some code, other people also drop everything they were doing and try to contribute, that is, to write and commit their solutions.

They don't do it on their initiative, it's the managers that tell them to go fix this or that file. But in this case we had two volunteers to come over and help me, uninvited.

Ok, I should not care, and I don't.

At the second meeting Eric was happily reporting his victory (he added an annotation, five line of code, and inserted it everywhere he could, that is, into the code I've been working on). That's in addition to two already existing annotations to each member of a class. I asked him whether we can create just one annotation, and insert one line? But now David interfered, saying that, no shit, the more annotations a variable has, the better.

Now tell me, is there any point arguing? I can tell them that it's always the less the better (one is enough), but what next? And, generally, I HATE arguing, I just HATE it. When you argue, you always feel like you are attempting to convert an idiot into an expert, which is, of course, impossible.

Meanwhile, Bill started reviewing the second version of my pull request. Now he started giving me wise advices on how to name a variable, and into which tables should the substructures be inserted (what t.f. tables, they are all in the same table, and the structures are JSONs, as the Jira case specifies).

So, I had tweaked it a little bit, sent it to him, and meanwhile I wrote a pull request to just clean the infrastructure. Like """{"what":{"id": 42, "name": "Ivanov"} }""".as[MyRecord], this kind.

And I sent it to Stephen, who is more competent in coding. Meanwhile, Arie is asking me what do I think about Eric's idea. I answer that it's a good idea, makes more sense than what we have now. But it does not remove the necessity to check the structures for isomorphism. So, we decided to have a meeting on this.

But spontaneous meetings in our company happen like this. You sit, you work, at a break you take a look at Slack: and there you already have a meeting, "right now". Guys, eh. People are busy, don't do that synchronous pinging. I read Slack in an async mode, when I have nothing else to do (when tests run). So I just miss a half of these spontaneous meetings, and they discuss things without my presence, and agree on everything. It's unlike planned meetings, which are in my calendar, and they ping me on time.

Oh, whatever. I'm so tired of all this...

February 08, 2022

Oh. All the time, an emergency and a rush.

My pull request, I had to restore it again. Bill approved it, I merged it.

Then there was a buzz, some endpoint does not return anything from the database (were the fields renamed?), and Bo shows on his Frontend: {"data": [{} , {} , {} , {} , {} , {} , {} , {} , {} , {}]};

I was asked to investigate, and I started investigating, but I need a database access, and it's not clear where do I get one. Wrote a test for that. Found that I need some access keys; wrote to Eric, asking for the keys, and we went for a walk.

When I returned, there was a noise regarding my pull request, I had a typo in an attribute name, so everything crashes (and who asked them to deploy? I was planning to add more tests.)

In short, I wrote one more pull request, but there was nobody to review it, so I'm deploying it as is. Firefighting!

Oh, Eric wrote how to access the database; I'll take a loot tomorrow, I'm too tired today.

Ha, not that simple. Life goes on. Bill called: he had added, into the same class that I deployed, some "type tags", and he's saying that nothing will work, and he will kick out his stuff and will redeploy. Promised me to call again. And he lives in New York area, and it's almost 9pm. (Is it still evening? Is it night?)

February 09, 2022

I'm pretty tired of all this. Combining trips to furniture stores with meetings over the phone internet, from a car. The meetings are totally getting me, killing me.

Actually, in the morning I started hacking something (last night Bill has finally commented out this new attribute, because it's not present in the database); at 10am our time Bill and Stephen showed up on Zoom, and we started discussing the opportunity of automating the database updates based on the data structure we have in the code. Bill has the brightest idea: as soon as we start the server, we update the tables, if necessary. Looks good. But for that we need to know all the attributes and their attributes; currently all this is written manually in a Python code.

We agreed to fix it. Seems like a new era is coming: "Juan knows Scala, let him do it". I have now tons of that stuff: go ahead, do it. Well, I do it, I love coding. I don't like the meetings. After this 1.5 hour discussion I need either a glass of vodka or to jump into the pool, or just lay down and take a nap.

Instead, we drove to Hickory, it's a kind of American Furniture Capital, they make everything there. An hour drive from our home (and a 1.5 hour drive back).

It's okay, but there was another meeting, and I did it on the road, and then when parked by the store. They also threw another case to me, in addition to what I was working on, also an urgent one. And since I was also working on yet another case, they threw it to Stephen. Stephen found a miracle cure to a bug: roll back my pull request (which was tested by the tests), and kaboom, everything works. This is a miracle! I'll experiment during the weekend, wtf was it.

Okay... then it occurred to me that fuck, why do we need a strict synchronization? If an attribute is not present in the database, we don't have to return it, except in the case when it's specified as "required".

Okay, whatever. Let me deploy it now, and I'll play with it. Does not look like a big deal. There's a question now, should I store the substructure as JSON, or split it into its leaf elements. I'd rather store it as JSON, but... we will see.

February 10, 2022

Mostly dedicated my time today to visiting furniture stores with my wife, but it's time to work, to do something, and I'm tired.

At least there were not so many meetings today.

Yesterday I added, manually, a column to a table in the database... in one of the "projects", but there were questions. In the morning I had some sex with the tests, and so far so good, but not without problems. Too boring to explain; it's routine.

At the meeting I asked a question: guys, are you sure you want nested structures in the tables, or we can just stuff JSON into columns; otherwise how are you supposed to refer internal fields, via a dot? And in the filters? Everybody suddenly agreed that JSON would make things easier. Actually, I know a better solution (flat properties, according to the core SQL ideology), but it's useless to explain it to them.

And they also are telling me that hey, there's a bug. Interesting; last night I was asking Bill about this bug. He had it pending for a couple of weeks. Now they threw it at me, because Bill is busy. And this bug, they say, has the same level of importance. Ok, I said, never mind, I'll work on it in parallel.

And lo and behold, they woke up, and started discussing the boolean "flag" that they added to the database: "data are being processed" or "data are processed". This flag will be updated, in parallel, from the browsers, dozens or hundreds of users, and also some scripts. Funny, right? So, they say, there is such a problem called "synchronization problem". And I'm telling them: do you remember I was talking about this flag already, that we have to take it more seriously, and, maybe, use a queue instead. Ben, thank God, woke up and said: "But of course!" All others are sitting quietly, and, I guess, our PM Sridhar will be discussing it with Prasanna, how we should synchronize operations (nobody seems to have heard about transactions). Or they give it to Eric. Eric has nothing else to do, right? Right?

Ok, so I sit, and work on some case, in parallel... at Google I used to run four tasks in parallel, via round robin).

We were supposed to have a 1-1 with Arie, at 5pm, but he's not there. Then he writes: "I'm in a meeting with Prasanna, sorry". Half an hour later he shows up, with, obviously, a red ass after spanking, and asks me: "Do you know this epic? Are you aware that it's the most important? When will it be ready? Can you finish it today?"

This epic contains eight cases, five mine, three are owned by Jordan. One case is: “add a field, making sure that it exists” (it was added, and the code is supposed to read it); the second case is: “implement editing of this field” (that’s what I suggested to them a couple of weeks ago, that the data make no sense if we can’t set its values, since when the record is created, the field is empty, and it reflects the status of a process).

The other cases are “write unit tests” (they think that I first will write the code and deploy it to production, and then I’ll work slowly on its tests), and write a documentation (I actually write documentation first, otherwise how do we know what we are implementing?)

What’s funny, there are three of us on the backend side, it turns out: Stephen, Bill, and me. We co-operate smoothly, three peaceful white engineers, no emotions, nothing. But why then it’s me who’s supposed to write this half a dozen cases. I have no problems with Stephen’s qualifications, he’s perfectly okay, but a little bit scared or nervous. As to Bill, he doesn’t care, he’s an ACM star, and nothing bothers him. Maybe that’s why working with him is a pleasure.

So, right now, as “it’s urgent”, have refactored (mostly cutting into separate classes) his 2.5 lines of code. Because I don’t like editing huge files like this. And I’m merging now, incessantly, I have no time to slow down.

But. I started at 8am, and now it’s 8pm, and? See, I’m deploying it now, then I’ll ping the end-to-end tests, and if the tricks (that is, experiments) succeed, I’ll legally add this field into the code. And will, in the next step, write tests for it.

You know why we are in such a trouble? The only legal method of updating the database is: you take a huge Python file, and manually add something there, with attributes, whistles and bells. Then you have to run it somewhere, to make sure it updates the database on Dev. Will it be compatible with the code? No reason to expect it.

In short, you see, we are building, in our circus, a rocket on which our circus bears will fly to space. Like Musk. By the way, Musk is also in the circus today, giving a speech about his rocket, 48 minutes from now. Google it out.

February 11, 2022

I feel like I work way too much; and it’s not obvious whether anybody will appreciate it, except maybe a couple of my colleagues. It’s also not obvious whether I need it at all. And more, at times the level of rudeness exceeds all norms. At the first meeting today Prasanna suddenly asks me what am I working on. I try to tell that I’m fixing the code, to make sure it does not nervously crash if a field is missing in the database, but just ignore it in the case the field is marked as optional. And Prasanna answers: you what, didn’t listen to Eric? He had invented, ten days ago, annotations for all this, and everything should be working now.

Oh, really? Eric has added some annotations into our Entity, a couple of days ago, but he didn't tell anybody, what is it for and how to use it. Now Eric jumps up and says that it's me who has to add annotations to all the fields and all the structures. And why exactly? That's when everything will be working, - says he.

Hmm, ok, why not.

Then Eric had sent me, on Slack, a tricky line that I'm supposed to insert in tests.

In short, it's been an hour now, I'm still inserting these fucking annotations.

It doesn't work. Eric has a bug. See, I'm the first one to try his shit (which has been, supposedly, ready since ten days ago). I showed him his bug, added a test case for his code, fixed the diagnostics. As a result, Eric, not even looking into it, approved my code, and fixed his (and I fixed it too, in the code he approved). OMFG> Will it work? Eric later asked me, privately (we communicate with him privately, so that management does not know). I checked it. No, it doesn't work. His serialization produces shit.

No, there was more. Bill had added a field of type [String, AnyRef] - and how, do you think, it's supposed to be serialized/deserialized? And what's the point? Well, okay, I've just fixed this code of Bill, no big deal.

But it doesn't work.

And we have meetings all the time, one after another. At the last one Arie was strangely too rude. He says, "Juan, last time you said that you don't know what to do next, now listen carefully, and if you don't get something, ask".

Wos, what the fuck, right? What am I supposed to do... I know what I have to do. At times I don't know what kind of new shit these people have just produced. E.g. "Enrich the structure with the new data". And all this nonsense is called Scrum.

By the way, I really absolutely don't care what some people somewhere there think about me. I never cared, but now even more (or is it even less?)

And, of course, Frontend people have scheduled, for the next week, to come up with a solution for the "synchronization problem". In a distributed system, where users randomly click the buttons, "Data Ready" and "Data Not Ready", which state is reflected in one bit of one record. Good boys! (There are no girls, unfortunately). I decided not to interfere with the idea of e.g. queues; Ben was also silent.

All that happened because this "data ready" bit was invented by Prasanna, and since he had graduated from Stanford, he knows all these sciences better than the rest of the humankind.

Okay, oka, and there's also une funny thing, a bug that a QA threw to me. This bug was not confirmed by my unit tests, so the QA sent me a URL and a JSON file, where it is confirmed, and I now managed to reproduce it, and so, if I didn't have to wait for Eric, I would have fixed it right away.

February 12, 2022

See, it's Saturday, but everyone's on alert.

Prasanna is writing me something like "when will it be ready", and the like. Well, I was planning to have it ready> But something doesn't work. On Eric's side. Eric's in negation, writing "ask the channel on Slack". Hmm, interesting.

Of course it's all shit. Bugs. I can investigate it, but it takes time.

Arie, on the other hand, suggests to document my actions. That's what I'll do tomorrow (Sunday?!)

Meanwhile, I continue testing all kinds of shit; just discovered that it breaks in the database service, where the guys (Eric, I guess) are reshuffling the data, mysteriously renaming the fields. In short, yes, I can reproduce the bug reported by our QA, but the bug is in the database service.

Not that I'm suffering from all this; I'm just curious.

Bill and Stephen are also online. Bill's father just died. But he's still online. Well, one can understand him; he wants to get distracted from all that grief. I don't think I could have. With Stephen we discussed the movie Das Boot.

Actually, I don't have much left, to code what I promised. Tomorrow. I was too busy. And I'm tired. Sorry.

February 14, 2022

This morning I a) sent my pull request to Stephen and Bill, and b) marked my Jira case as "under review".

Then, on Saturday, I wrote to our leaders: here's my test case, here are the logs; keys are being renamed in the database.

But later today Eric answered, showing a UI picture, where they are not being renamed. Wow, oops. Okay, - I wrote to him, - I'll investigate. (Fuck, I'm writing it right now, and it seems like I got it how it happens (or not).)

By the evening Arie suddenly signals: Juan, Eric just told me that your `get` and `list` requests don't work. What, where, - I answer. Eric writes that it's about your question regarding renaming the keys.

I answered to Arie: false alarm; and as to the list request failure, I'd like to listen to the witnesses. Silence from Arie.

Otherwise, we had two meetings. First, it's our small team; second, our large team. Well... it was mostly the Frontend people discussing their problems, and I'm just sitting listening waiting when my

name is mentioned. If they mention it, I'll answer. Otherwise, interrupting their scientific discussion, how they form their forms, there's no point in that.

In the evening, while debugging, I had refactored a huge piece of shit that we had in our code. As soon as I created a pull request, I see a message from Stephen, saying: take a look at my pull request. Ok, I took a look. Funny, he did the same as I did, but he cut it better into chunks than me. Also, he wrote a better mock database, turned my mock into something more universal. So I wrote to him, ok, I won't merge my stuff for a while, go ahead, do it first, and I may improve it a little bit, but everything's fine, and I approved it. He postponed it till tomorrow, he's also on the East Coast.

At least some relief.

It's about 9pm now, and I'm still working, trying to trace that bug. There was a bug, thanks to Bill and, partly, to myself (I failed to trace this problem, oops.)

February 15, 2022

I got stuck today, really stuck.

Spent all day hacking, implementing `equals` and `hashCode` in stupid data-class, because nobody bothered, and now I'll have to compare instances in the tests.

But by the evening another shit happened. QAs and PMs started complaining that something broke on Staging. Well, what broke. They didn't update the database, but they "cherry-picked" the code. Now what, now it doesn't work with the old database. And to update the database, a cherry-pick is not enough, the whole module is needed.

Arie in panic, writes to me, asking, Juan, explain it how it could happen. I did explain. Explained to Arie, explained to Santosh the QA, explained to Sridhar the PM. No need to explain to the Frontend people: they are themselves literate enough.

I listed to them four options what we have:

- 1) Manually fix the database, in each "project". It will look especially good in Production, when it crashes;
- 2) Update a Python script (which I don't know and nobody wants to look into)
- 3) kick out this "substructure field" for a while
- 4) take what works on Dev and is actually in master branch on github, and deploy it to Staging.

The people were running around in circles, in panic, until Prasanna showed up on Slac and said: Number Four. Good! Thank God! We are for the progress.

Well, and since I got a lot of tests already, I ran them, and found how to extract *correctly* the data from the database (Eric did document it yesterday, I read it today; before that it was an unknown science).

Ok now. I deployed it. Fuck, it doesn't work. Broken. I let them know. The same code worked in the tests, but on the Dev server it doesn't. Well, okay, they have 6pm over there in California, so I still have about 14 hours.

What can I do. I added a bunch of alternatives, deploying it now. It's almost 9pm here. There's an interesting night ahead. I must fix it. Ok, I will. Strange that Eric's code works locally, but not on the server. I'm getting a feeling that on the server the library is different and the config is different. This is an interesting news.

Ok, increasing the vigilance. I love it, panic and firefighting.

February 16, 2022

Oh. Last night I was working till 11pm, but eventually have managed to solve the problem with get: and what seemed to be a problem, eh? See, they had a bunch of parameters for this get request, something like 9 parameters in addition to the id. And it doesn't work. So I used an alternative trick: search with a filter, and select the right record in the returned list (there are really not so many of them, after all).

Yesterday I sent a “delayed message” to Eric, on Slack, and went to bed. In the morning Eric writes that he's busy now, let's do it after 10 (Pacific time). Ok, NP. At least it works for me. Then I wrote to our QA; he verified it; now we can push it to Staging.

After that I've slapped together a cherry pick pull request; lots of stuff had to go in there, because of dependencies. Github says:

“Staging is ahead of master by 260 merges” - isn't it beautiful? Are we working or are we just having fun?

Then I sent this cherry pick to Eric for review (that's the rule, for Staging.) Eric writes: “too many files, ask also Stephen and Bill”. So I did. I asked them. Now I sit on my couch, doing nothing.

Meanwhile, Staging is in a state that does not allow our Frontend people to work. Not bad, eh.

At the meeting at 1pm (it's 10am their time) we had lots of scandal, like, how come that our code is not working, it blocks the testers and the Frontend people. And it seems like I'm guilty. I tell them: I didn't deploy anything to Staging. Really, I didn't. So, whose fault is it? Frontend? Me? Everything works for me on Dev (I fixed it last night, after it broke yesterday afternoon, suddenly, after a rather innocent merge of mine).

So there.

And nobody is reviewing my code. The next meeting is dedicated to the “data ready for export” flag. This boolean flag is inside a substructure, so we will see how are they going to “update” it. Like, we

read a record, set the flag, write the record. Do it in parallel, on the servers, in the scripts, on the Frontend.

Prasanna spent about an hour explaining when this bit should be set; during this process I was writing a short document regarding that. As soon as I asked, when do we clear this flag? Prasanna said that he has another meeting, and disappeared. The others, of course, have no clue, when this flag is cleared, but it's considered unimportant. Project Managers are ready to create Jira cases. Me too, I had my next meeting, with Eric, Bill, Stephen. They are already discussing the "automated database update". Our database is not a one single thing, but it's per project. And now, it turns out, there's a new code recently added, "init", that, on the server start, tweaks the database according to the new class structure. Then they were talking about how important it is that class annotations are correct. We have rules, etc. Well, in my view, if you have rules, maybe we can generate these annotations? Well, arguing with some people may cost you. With a difficulty, I have extracted from Eric and Bill answers to two questions:

1. You were saying that everything is automatically generated, but we need to add the class name to "init", right? Right? Eventually they admitted: yes, they all should be listed.
2. If the project is added after deployment, who is going to update a database for it, or who is going to build the database? Now they basically convinced me that this is not important. So I ended up writing to Arie regarding our status of automation. Then I started getting on their nerves: guys, are we still going to continue blocking Frontend testing on Staging, or you approve my cherry pick pull request? And that's where they started criticizing my code: How come so many files? Where are your annotations? What is this line? What is that line? And I tell Eric: "get method does not work in my case." Why so? I don't know, Eric. See. And I show to him that when I send a request, I get a record where only "name" field is filled.

Here Eric suddenly says: send me 'curl' and 'json', what are you sending to my service. And he runs away: he has a meeting.

Now, what do I send him, I send nothing, I call his method in his class, I don't send anything. Why does his method need an instance of Request from me, Anyway, it's incurable. Nothing's curable here.

Then I wrote to Arie that we are delaying the deployment, and the QAs won't have time to test it on Staging, and we probably should cancel this new field that I added, or else Production will crash Monday, all Production, when they start deploying from Staging.

I was extremely exhausted, no moral or physical energy left. So I closed my work computer. But before that Stephen had sent me a list of annotations, where I wrote field names not according to the rules (as if it matters, it is just a list of synonyms, we could do without, and Eric says the same).

Whatever... Tomorrow I'll kick out my cherry pick pull request. Meanwhile Bill and Stephen decided to check and improve my annotation. Good luck, I greet the initiative.

I've really had enough all this, especially when the market jumps up so high, as today, so that the

feeling is: fuck all that job, just stay home and play with stocks. (Note from 2026: sure; that's what I do now, earning more than at this stupid job.)

February 17, 2022

The circus has arrived!

Last night, with a significant mental effort, I had closed my work computer. Now, still while in bed, at 9am local time, decided to take look, opened my work computer again, to check it out, what's up.

Well, nobody touched my cherry pick, nobody approved my pull request, and I've lost a temptation to merge it and go ahead break the production. I wrote a long epistola (seven lines) to Arie, explaining what's up, and then another epistola to the team, explaining the nature of things.

What was next. Stephen and Bill have carefully fixed my typos: the property name that I wrote in annotation as parametersOfSentinelRun, which sounds like "Parameters of Sentinel Run", they replaced it with "Parameters Of Sentinel Run", convincing me that otherwise nothing would work, and everything would break (my tests do pass, but these two wise men don't seem to trust the tests, but they trust their gut feelings and their own being always right).

Do I care? I do not, and I just waited for them to settle and to approve my cherry pick, but the longer I waited, the more I understood that this cherry pick should be thrown away, and I have to kick out this data column (with a large structure inside) that I had added (by mistake?), and we need first to fix all their stuff and test it properly (and there are many skeletons in their cupboards...) Then we had a "standup", where the Frontend people were discussing their forms and asked me when I'll fix, at last, the "business unit list" on staging.

What's good, while these Frontend people were buzzing, I'd assembled two bookcases in my library.

When it was eventually my turn, I told them what options we have. Either we kick out this column for a while, or approve my cherry pick. But a) there's no guarantee, and b) Frontend have just a day and a half left for testing. And then Eric jumps in and asks, whose idea was it to write such a huge cherry pick update? Well, - I said, - it was one of the four options that we discussed two days ago, and Prasanna chose #4, "Bang, bang — and straight into production, all that we've slapped together". (And then he'll be screaming that oi-vei, nothing works in your code. I did read Feynman's book about Shuttle, and you did too, and everyone did. So, at that meeting I was asking everyone, what they prefer. Arie was silent. He's not Prasanna, he's not entitled to an opinion, he asks others. (Which is pretty wise, no?)

Then Eric asks, why can't we add a column in the Staging database. We can, - said I, - but then Production will crash, since it's not there, and we'll have to add this column into all projects. We'll do i, - says Eric. Well, ok, then everything's fine!

Seems like I've convinced them that first we have to make it work, and then beautify it to our taste.

After the meeting I took a break, had a cup of tea, and went ahead to check, how exactly was I adding a column in Dev.

See, we have two variants. Either you define the data column as a structure, manually listing all the fields, which are also structures, and so on. It will take about half a day of manual work. For each project. Or (my cheap solution) define it as a string, and put a JSON there. Because a) Postgres understands JSON, since long ago; b) nobody cares about individual fields (there's a really huge amount of work, to write an API for updating individual fields in substructures, nobody did it yet, and nobody will ever listen to me regarding "Properties Are Flat".)

In short, I added a string in Staging database, pulled our QA to check, he checked it, - wow, fine, we can breathe. I took a long break, after having reported the progress.

Arie, as someone responsible for updating the database in Production (nice people are always responsible for everything), he was asking me the name of the entity. Well... it has a different name in Production... and so it goes.

Then we took a walk, me and my wife, and when we returned, Slack was full of screams: "Juan, your bu/list is throwing an exception." What exception? It was working on Dev in the morning, and after lunch it was working on Staging. Now an exception.

It was a stupid exception, really. See, I take the first character of a property name, and check whether it's a capital letter or small, to know what to complain about. And oops, the name is empty. Property name is empty. (Remember my dream about a bug in a cat's name? That was about this.) Seriously, how can an attribute have an empty name. I ran one end-to-end test, checkout the logs on Dev: wow, there are tons (a list, a set) of names that are empty. A set of attribute names of a record. Nine attributes have empty names. What happened?

Now there's a panic. Bill decided to fix it; he's asking me, what does this exception mean (`EmptyIteratorException`)? I write to him: no worries, give be 30 minutes, I'll fix it. And I write to Bo, our Swedish Frontend guy: no worries, give me 30 minutes.

And as soon as I found where all this comes from (while I went for a walk, Eric and Bill stuffed some "improvements" into my code, using the new Eric's code, which has zero tests, there's a Good News on Slack: Eric has fixed the exception. Juan had an exception, Eric fixed it.

As soon as I decided to write to Eric, asking to ping me when he deploys (they are clueless about tests), I get a message from Bill, saying: hurray, I've deployed Eric's code.

Now I run my end-to-end tests. NPE on the server side. It was a beautiful fix.

So I wrote to Eric, asking whether we first ask Eric and Bill to roll back their shit, and then, tomorrow, first test their code locally, before rolling it out to the servers. Arie: sure, go ahead!

So I write to the Engineering Channel, saying, hey, Eric, Bill, please roll back your changes that caused the failure(s), we'll take a closer look tomorrow, but let's not block the frontend people.

And now I turn off my work computer, because it's 8pm here, and I have no intention to start discussions with Bill, and there's no point in discussing anything with Eric.

But before I turned it off, I see a message from Eric, asking me to send him "my curl call that updates the database". Wait, Eric, who, do you think wrote all this code, not you? We are updating the database, these days, manually, via UI, and I already did post, a couple of hours ago, how to do it. And I have no clue how to do curl to your API to update the database. Actually, I know about this API of theirs, but it's not working anyway, and starting discussions of this kind after 8pm is kind of stupid.

So I thank and took my leave. I better watch a French movie, it's Nice in that movie... and no, I don't like Nice. Nice reminds me San Francisco, too much.

February 18, 2022

Too many activities today.

In the morning it turned out that nobody rolled back anything; the progress progresses. Brave Eric writes to me: "Juan, tests on dev are failing, go fix it". I answer: "Okay, if you want me to fix what you repaired, create a Jira case, I'll work on it."

And then I write to Bill, asking whether they are going to fix what they didn't fix yesterday. Bill answers: "I'm going to my father's funeral, he had died several days ago". I already knew about it. He attaches his father's pic: his father died when he died at 82. So I write to him: "Oh, sorry then, never mind, I'll take care of all this", and expressed more condolences. And then I started thinking, wow, how old is Bill? He looks 75, talks slowly, repeating, you can't interrupt him with a question or explain to him that you already know these trivial things. If he starts his discourse, I have to just open sudoku on the phone and wait for the end of his speech.

Oh, what can we do. Wrote to Arie, just to let him know.

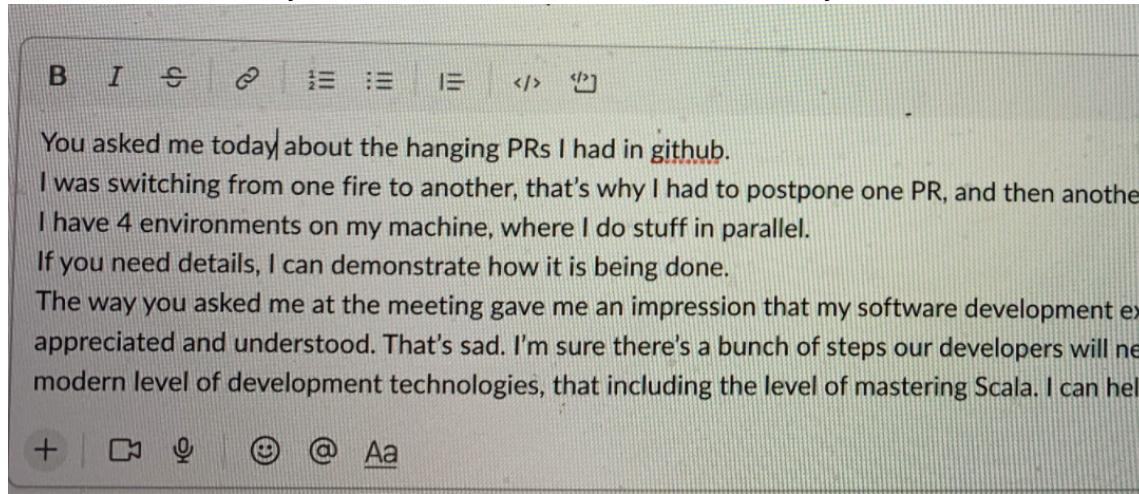
As to rolling back, Prasanna too writes that there's no reason to roll back, we have to fix what's broken. Eric, meanwhile, repeats that his stuff is not broken, it's me who has bugs in the code.

Then the meetings started (it's Friday). The first meeting is "Demo". But nobody demonstrates anything, and Prasanna goes on with his non sequitur speech, and then he starts interrogating the people. He asks me what I'm busy with. Well, I respond, first I was fixing the broken Staging, now I'll be fixing the broken Dev. And also, there are some serious questions, like, how are we supposed to pass around updates of nested structures? We don't have anything for that, I plan to design it next week. What's there to design, - says he, - we already have a solution, pass a path and a value. Hmm, thank you, - say I, - I have not seen this old solution implemented yet, but I was planning to do the same thing.

(Actually I did not expect the team to even understand this solution.) And he says: it will take four hours, not a week. Wait, who told him it's a week? - And what else are you working on? - This week I'm spending all day at the meetings. This doesn't bring money, - says he. Wow, my point exactly.

And now he says: "you have this week-old pull request on github, why?" (He doesn't know that some pull request here are six-month-old, he doesn't see them). Fuck. I don't want to discuss it, - I said, - I'll take a look.

Actually, I got insulted. I don't like it when they are rude (I have to confess, I was like this myself). So I



wrote to him on Slack:

He writes: "I know you are smart, that's why we hired you" (Which is rather insulting, actually), Then he says that I'm working too slow (which is also bullshit, in the majority of cases I roll out everything on time, and besides, I'm fixing things, silently). And he says that it's a startup, and we work fast. (I see how fast it is, they talk fast, but it takes a month to implement a simple feature.) And in the end, according to his books, he added something positive.

I wrote to him that I adore the startup style (yes, I do), and let's accelerate development productivity. E.g. Eric showed how he deploys locally, but nobody can reproduce. (It's simple: Eric has access to everything, and we don't, so we can't retrieve keys, configs, etc, nothing.)

And he says: "Seb already showed it, there's a video."

I know that "programming by video" style. It's usually not reproducible.

In the end we parted, both happy about ourselves and the seemingly achieved mutual understanding. I don't know if it will help. I'm really tired of all these slow processes. I can only work on weekends, when nobody is jamming the servers, the deployment, etc.

Then immediately we had a "standup", where we figured out pretty fast what problems we have and in which order we are going to fix it - but then Prasanna showed up (let me remind you, he's a CEO), and started teaching us how to develop software.

Then we had another meeting, and Prasanna was giving an 1.5 hour speech, telling us how everything's fine in the company, in particular, with the money. Now I'm getting my doubts.

Then there was a break, and I could drive to the pharmacy to pick up my stuff.

And then the lat meeting, just an hour and a half, "scrum planning". We discussed again the priorities, the future plans; everything was quiet and businessline. But then Prasanna showed up and started teaching us to develop software, and asking us why this and that is not done yet. Opened one case, which is assigned to me; in this case we need to give everybody an access to the list of business units. I think I did this some time ago, and I could remember. But wait, I need to fix Dev, no? Priorities. Then Ben pops up and says: "let me fix it". I said: "thanks, man!". The case was reassigned to Ben, and then, after the meeting, Ben started asking me, can I show him in the logs what's going on. I sent him a log from Dev, where this exception happens. It seems like it's time to write a script to view logs live, filtering out the garbage, of course. Too much garbage.

Meanwhile, while they are talking, I was working on that broken build. Bill and Eric have added a bunch of interesting stuff... it's kind of good, but why didn't they finish it?

Well, who cares. This weekend, in my free time, I'll look into it. While nobody interferes. See, I got sworn statements from the colleagues (e.g. from Stephen) that they won't come to fix the same class. I'm really tired when they, in a big crowd, start editing the same couple of files, competing who's the first to commit. Of course that's the company culture. In the monarchy it's usual that the servant rush to fulfil the monarch's will. And who doesn't rush, he's a slow employee.

How it is good that I don't fucking care. Even if they kick me out, is it a problem? Can't I find another job? And then enjoy the honeymoon at that next job? I love it, these fucking honeymoons. And I don't like it when I'm not appreciated.

February 21, 2022

Yesterday I had figured out eventually, why t.f. one part of functionality works with this kind of annotations, and another with another kind. Eric needs one kind of annotations for the database, and we, to receive data from the frontend, need another kind. He doesn't care, of course.

So, I wrote a pull request (and merged it) yesterday, to experiment, and then to create a fix.

Meanwhile, also yesterday, Bo, a diligent Swedish engineer, tried the API again, and it doesn't work. Wrote about it on Slack. Eric shows up, saying: "looking". A couple of hours later Eric posts a sample curl command, saying: "see, it works". Interesting, I tried it. Yes, half of it works, half doesn't. Looked closer. It turned out, Eric rewrote lots and lots of code. Sure, nobody sees his rewritings; he, same as me now, merges on weekends, not showing it to anybody.

Now what. I'm not rolling back my merge; there is time. On the other hand, I don't even know whose merge came first, most probably mine, and Eric's, it came later.

I could fix it, but in our guerilla situation, what's the point. He'll break it again. Earlier I thought it was all Bill's. Now I know that it's Prasanna and his culture of a gang of guerillas. In principle, a gang of guerillas is not that bad, but somebody has to coordinate, and here everybody coordinates only with the chief, ignoring all others.

As a result, I have a couple of hours of free time. This reminds me the soup that the communists were making in "Chevengur" by Platonov.

February 22, 2022

This morning I wrote to Bill and Eric, asking to either coordinate our updates, or to confirm that they'll be fixing it, not me, or that I'll be fixing it, and not them. Having three people on the same bug in the same piece of code...

Bill wrote that he has no plans to touch this class today. Eric wrote that I have to contact Arie with such questions. Which, to me, demonstrates that they had got Eric involved to work on our code out of fear that we, Bill and me, will break something. So I sent this Eric's answer, together with my question, to Arie.

By the way, yesterday I set up our team's space on Slack on my phone. Now I can talk to them even faster than on the computer (where one has to click, to log in...) And now what, Slack has just crashed, for half a day, all around the world. Interesting, interesting, I like it.

The we, the backend people, had our first meeting, almost like a meetup. Bill, Stephen, me. We had a really great, impressive discussion. We discussed the history of all this nonsense. In particular, Prasanna gave Bill 24 hours to rename all the properties. So Bill wrote a special code for this: take a variable name in the code and convert it to a property name (why do we need it, nobody knows, except Prasanna).

And we have figured out that Eric's code converts a database record into JSON in the format that Eric wants. What format do the frontend people need, it's not Eric's problem. Nice, really nice. Now we know where the problem is, and we can fix it.

But later today we had a meeting, a "standup", and we had a long discussion again regarding this. Arie does not feel well after his surgery, and he'll have another soon. He's also much thinner these days.

And Arie suddenly writes to me, saying: "Please test your code before you deploy it." Wow, is he okay? Why doesn't he tell this to Eric? Is he afraid of Eric? And I answer: "Seriously? When I was fixing Eric's errors, he fixed one part and broke another..." Etc. And he writes: "There's too much broken, this method of working is not sustainable". And I answer, saying that Eric has 0 tests, and how come it's

my fault that he merges his code without any review or any tests? Ask Bill, he has something more to say about all this.

We touched this issue too, at our backend meeting. Now we are some kind of a gang of Scala developers.

Next there was a big team meeting, an Arie wasn't there (he took off for his next surgery), and Prasanna was bugging everybody. This time Eric had problems attacking me, because Bill has something to say, and Stephen too, and me too, so, in short, Eric backed up and started talking some kind of nonsense, like "I was trying hard". Not his usual "fix your bugs", which he was addressing to me.

So, this is a race condition. Two Chinese, three Whites, a couple of Indians on our side, Shridhar and Santosh, working with whom is a pleasure. So, so far I stay.

Have more indifference! (less difference)

By the way, David also broke our tests. He had change a generic format of error messages in one of the classes. Not bad. Well, that's the usual thing. But I had to fix the tests (generalized them).

So, now things are getting slightly better, all tests pass except a couple of end-to-end ones, but I'll work on them after the deployment.

Another thing is, the server has so-called "projects". And the end-to-end tests count on everything being fine in the project, but it's not so actually. For instance, this functionality, "update the database structure during deployment", the feature that Eric was so proud of, it just doesn't work. As a result, end-to-end tests pass on some "projects" and fail on the others. I'll have to investigate. Tomorrow. Let it get deployed now, and I'll close my work computer until tomorrow.

Aha. I did take time and ran the end-to-end tests. Update does not work (I have a vague idea why); but more, the logs don't seem to have my tags. Why so?

February 23, 2022

So, okay, at least they are not bugging me with asking "when will everything be ready?" (it will be ready when I fix Eric's improvements... well, Eric also promised to fix them, today, publicly, he promised to Stephen to fix handling of Options. That's how it all works here. Two serializers having the same name, one in one package, another one in another; one understands Option*i*, the other doesn't. And silence.

Okay, then Stephen called, we have to deal with my old class, where I did the serialization via JSON, but now Eric has introduced a "generic" solution, and we should better use it (hope it will work), so we discussed it.

Then we had a standup, and somehow they assign all that shit to me (and I am some kind of volunteer). In the end of the meeting Bo, a Swedish Frontend guy, complained that a) somebody is adding to the

test project (that's into the database) is adding Business Units (I do, in the tests), and now we have not two, but six Test Business Units (but I delete them in the end of the tests). So that now API returns the list of Business Units (six instead of two), and the list is unsorted. Prasanna already explained to the people that you can never extract a list using 'OrderBy': our database will hang when sorting two or six elements, because our database is distributed.

I asked Bo if they can, on the Frontend, add a line of code to sort the data they receive. Yes, we can, - he says.

Also promised to him to never again return test records, as if they never even existed (no big deal if you can).

Now I started working on this new stuff, and then they threw at me more new bugs. They are not exactly new, they are from the last summer, Bill wrote them. E.g. when we get a request for an empty list of Business Units, we return "Server Error" (the error is that, when running this request, we experience an NPE). Nobody, of course, checks whether the database (to be more precise, a thick layer between our service and the database) returns deleted records. The records that were returned. I have to, in my code, check whether the records that I deleted were actually deleted, and then retrieve the list of remaining ones. Because they return all the records, deleted or not.

This is crazy. Now, a) who will write tests for all that? b) Can't this thick layer, by default, ignore the deleted records? You must have seen the sequence of SQL statements that is executed for each select (a long list of fields, observed when I worked at "WeSell") from Entity where entitytype="Business Unit". Notice it, are all records stored in just one table, Entity?

Not that I care. We will see. The general slogan of the day today is:: "Juan will add a test case and he'll fix this bug".

February 24, 2022

Too many things happened today, it's hard to concentrate.

In the morning we had our backend meeting; Bill came, but Stephen didn't; and we discussed with Bill all those issues of data conversion, and where does everything come from. When you talk 1-1 with Bill, he abandons his "professor" style of talking to the masses, So, everything was fine.

Then what, then we had a standup, and there Arie informed me that this is a very urgent task, we need to do it urgently, and I'm spending a week working on it. I told him that new problems pop up all the time; yesterday I was working from 8am to 10pm. He got quiet.

Then we needed to talk about how to fix all that. I explained again what was the problem, in my view. And Arie set up a meeting, Eric, me and him.

Now we have a meeting. Eric says: “you fix your bugs, and everything will work”. What bugs? Eric says: “the type discriminant for a structure, you call it ‘Type’, and it must be ‘type’. Wait, is not it your code that transforms the data. No, he says. Wait, who’s done that? Bill improved it? Got Bill to the meeting. Bill says:”Yes, I made that mistake in the beginning...” Etc.

I was going to suggest to make our code a little bit smarter, to understand it both ways. But then the people got an even better idea: let’s ask Jordan to fix the data in the database using plain SQL, manually. Do I mind? If someone else must work, I don’t mind.

An hour later Jordan fixed it all, and we could, in principle, enable this feature; but I was already busy investigating, how come UPDATE is not working.

Then we had a 1-1 with Arie, he smiled and said: “ok, I see you are busy, we have to fix it ASAP, so, see you later”. And he disappeared.

I found the cause. We changed (they changed) the annotations, but the code was expecting the old annotations for the fields, and so everything was falling into a black hole, silently.

Now I fixed that, added tests (there were tests, but they were testing one class, the one that works, and what was broken was another class, and here you go.)

Now it’s getting deployed. It’s almost 9pm.

February 25, 2022

Oh, too much text today, many things have happened.

In the morning I noticed that one of the fields of the record with which I’m working is not getting serialized, because it’s a structure, but it’s a String in the database. I’d expect the library could guess about the data type, and convert it to JSON and back. But well, will do it later. We need an annotation here. Added the annotation.

After some extra tweaking, the tests started to pass. Except that, oops, I write it okay, but it reads garbage. How com?

Our Swedish Frontend guy, it drives him crazy, how come, one Business Unit is ok, another one returns a null. Where does the null come from? Well, it’s JSON formatting.

Of course we should never return any nulls, it’s a bug. Ok, fixed it now, and also, I’ve just banned this field, except for if a test is asking for it. And another field, I banned it too, because we have the same problem with tags.

Now everything works beautifully.

So, I showed this solution, disambiguation between a String and a structure, to Stephen, he needs it too. Eric is ok, but he does not understand our demands. Otherwise he’s ok.

Now I have to cherry-pick it; we agreed that since the backend on dev does not complain (it reports problems in the logs, but sends something beautiful to the client), so I need to cherry-pick. It's now after 10pm, and I'm still cherry-picking. About 30 files. Will send it to Stephen and Eric, let them read it, my job is done.

I'm really wondering about my status. "Juan will fix it, Juan will explain it", like this. It's been a long time since I was in such a position.

Oh, also, at the last meeting (we had three, summing up to five hours) they started asking again, whether we might have a race condition if we use just one bit to signal "data are ready". I interfered, told them that one can't implement concurrency via just one bit in a database, and it should be designed professionally, and since Prasanna said that one bit is enough, we need to get details from him, how can we use one bit for synchronizing dozens of processes, some set the bit, some clear it. A mystery, right? In the old days I'd be boiling, but now I just smile politely. Is not it a progress?

Now it's almost 11pm, and I've slapped together a cherry-pick pull request for staging, sent it to Eric, sent it to Bill, sent it to Stephen, for review. Let them enjoy it.

Stephen has similar problems with updates. (Fuck, even though Eric is a talent...) I'll try to help Stephen tomorrow. BTW, tomorrow is Saturday.

February 28, 2022

Just merged a week-old pul request. It was seemingly an important case, but it was postponed. The solution consisted of five lines. Size does not matter, right?

Then there was a standup, and I explained twice the status: what's on staging, what works, what does not and is blocked, and how are end-to-end tests doing.

Next a short break, and another meeting, 1.5 hours again. Mostly they discussed strange new features in strange terminology. Intake service. Is it the same as NGINX? I have no clue. But they did not bug me, so I was reading tg, twitter, etc, listening to the new terms they invented during the weekend.

Now I wonder, does our company even have a CTO? A CIO?

Ok, just looked. No, our company does not have a CTO. Or a CIO. The CEO solves all this problems. That's why he's spending time with developers telling them to use one bit for synchronization and asking how come a task that should take 4 hours takes a week. As if it's not his responsibility, as a CTO, to eliminate impediments.

So, I'm pessimistic, but I don't fucking care, I just enjoy it.

So, prepared another pr, cleaning the work area, simplifying some stuff, so that next I'll be restoring, repairing, and ensuring that they work, a couple of features that I had blocked for staging/production. It was reviewed, and I merged it.

The reviewer, Stephen, wrote to me that wow, he was working on a similar PR. I wrote to him that we should coordinate. He sent me a PR, 32 files. Beautiful code (just a little bit too verbose). I approved it. But then I remembered Tom Chan from Salesforce, a meticulous guy, from Singapore; he did run tests on the code he was asked to review. So i did. Half a dozen tests failed. End-to-end ones. Wrote to Stephen, then we chatted on Zoom, he said he'll fix it all tomorrow, it was too late (almost 8pm on the East Coast, and he's here too).

Ok, anyway. Wrote to Arie how great is Stephen's code and how good are his solutions. I had to express my emotions, but it's impossible to do it publicly. Failing tests? He'll fix them soon. It's not worth mentioning. Good that Bill is working on "Intake Services" now. Not interfering with his shit. Good.

March 2022

March 01, 2022

This morning we talked with Stephen regarding what to do with his huge PR. And I promised to work on a specific test suite - it was about the code I wrote, and nothing was failing before refactoring.

Actually, the problem was about interaction with an existent server, so it has the right to crash - but I added some more investigation in the code, and merged it into Stephen's branch.

Stephen then added more, and kaboom, tests started failing.

Eventually, after lunch, we did all this together, our attempts to merge. The problem was though that by the moment he wanted to do `git origin merge`, master was updated A LOT, a fucking lot, so it was useless. So eventually he created another PR, with the same code (I do it every other time), and it was successfully merged.

Then I ran the tests, investigated the behavior; civilized some tests again, sent for review, wrote to the team about the status.

Well, before that, we had a standup, and then again I had to explain what's going on, and actually get something like a vote and an approval from our PM for Stephen's huge PR. Used the right words ("move on fast", "Prasanna", "Prasanna's solution", etc).

That's it for today. Actually, being under stress, I did not work much. Just as an interruption from browsing and chatting with Ukrainian friends here and there.

March 02, 2022

In the morning spent some time fixing the funny bug where the "transformer from field names to prop names" was being called twice, with a stupid effect; made it idempotent, merged, deployed. A little bit less bugs now.

At the standup meeting we discussed again the issue, why "update Business Unit" has been broken for about 3 weeks. Well, each week I fix it, and then people come over and improve it. I think I'll have to talk to our PM regarding how we should stop this behavior and somehow enforce TESTING of the code they commit. Or just leave it all to me. And MAKE THEM send the code for review to me. Since I'm responsible for this shit anyway. Will talk to him tomorrow.

Meanwhile, fixed the behavior for Business Unit. Showed the fix to Stephen (it's generic, just inherit the right trait); he's struggling with his things, same shit.

Merged, built, deployed. Deployment failed. It was already dark outside, we were sitting by the fire, drinking wine, and I was working. Anyway, wrote to our "server" guys, asking what's next and what to do about "deployment failed" message.

March 03, 2022

Ok, my yesterday's attempt has failed. The interpretation of the data in a request depends on what request is it. Oh fuck.

Added experiments to the server, and started just calling it this way, and if it fails, call it another way.

No, first I looked through the server logs... I'm on the verge of adding to end-to-end tests a trick of pulling out the logs and their analysis. So far I just have a script for that. I can do tail -f, and send the output back to the testing machine. Or not.

Ok, so, I was ready to fix it. Bit then, kaboom, Stephen pops up with "I've fixed something incidentally". No, actually he sent me a pull request, but have not we just agreed that he won't be touching this class for a while? Well, whatever.

At the meetup I solemnly promised to finish this feature within two hours; actually spent 45 minutes on it.

I already have two places where the server communicates with the database like this: "let's try this method, if I get garbage, I'll try that method". Now some kind of stability settles, and later maybe I'll be using the one that always works. It wasn't obvious, and you can't get anything from Eric, he has his standard answer: "fix your code, you have a bug". Fuck him, this Eric. Often I just copy his method to my class, tweak it, fix it, and it works for me. Stephen too complains about this (and he has things to fix), and I advised him the same trick. Later we'll clean our code, kicking out the trash. When all the functionality is ready and is covered with tests. Otherwise, go figure, send it to a QA to validate, or, worse, to the Frontend people.

As soon as I deployed this stuff, and started turning on the end-to-end tests again (those that were turned off), Arie comes up with "Vlad, while you are at it, fix also this case". Well, it's not fixing, it's just adding a field. No big deal, added the field.

It's probably time to start a good refactoring. Introduce meaningful types. And avoid Eric's "libraries", which have zero test coverage.

Whatever, it's nothing. Sent the pull request, Bill approved it, I merged it, and went out to take pine needles from the lawn into the woods. Should not I do something useful today?

March 04, 2022

Life is struggle today. In spite of back-to-back meetings. In short, everyone now started using this new... not a feature, but an opportunity about business_unit/update: and now it turns out that they update, but it doesn't get updated. To be more precise, it updates on my test records, but in the "main" records it doesn't get updated. But in the database, if you query it, everything seems to be fine.

Hmm... I think the server returns cached data. Right? Something from yesterday. Or from two days ago. See, if I create a new record, and edit it right away, it's considered as fresh in the cache. And that old record that we are editing, it's not reread, because it sits in a remote cache. Probably. Meaning, I just got it, what's happening.

Spent half an hour, wrote a test case that demonstrates this wrong behavior.

There are other things, like, I take a list of business units, get one (the first one) from the list, and “read it”. And the database is telling me: “no such thing”.

In the end of the day, as Arie suggested, I created a Jira case, assigned it to Eric, and he's now bugging me with questions like “did you try to read the list of units?” I did, and? Your code is not working, here are two test cases, they run the same test on two different records, it works for one but fails on another. He could thank me. But he, in his style, is not cooperating, pretending to be an important idiot.

Well, fuck him. Also, our Swedish guy Bo found a “wrong answer”: a client receives an error null with the message null. Bo just wanted to list some records. And now what? I wrote to Stephen (it's his part), then I took a look at the log. I see an NPE. `NullPointerException`. Which happens when we scan a list returned by the database (trying to extract something from the list). How come the database returns nulls?!

Haha, while describing it, now I know how it happened. The database reads one data format, and Eric's Code tries to convert it to something different. When it can't convert, it returns a null.

Most probably I have the same problem, but not with the whole record but with some (now I found two such cases) of the structured fields.

Here you go.

Prasanna writes, on this topic (he has nothing else to do, I guess), that we have to skip bad data. Data Fucking Scientist. A measurement error. It seems like he never dealt with databases. But at least he knows the word “transaction”.

Oh, fuck them. No need to try too much. I have filed a bug, and the rest is not my business. But how come they throw all that at me? Who knows. Maybe there's nobody else that could do anything. They also throw stuff at Stephen. Stephen is pretty literate, but he doesn't waste that huge amount of time on this nonsense, as I do. Oops. They won't thank me. Eric will complain thrice about me (my sarcasm), and they'll kick me out.

March 05, 2022

Okay, it's Saturday, but I could not hold my patience and took a look at Slack, what are they saying?

Eric, in response to the bug I filed in Jira writes, hey, write which curl command do you use, and what data do you have and then I (maybe) will investigate your problem.

I think, while it's rude, the idea is reasonable: a test must report a behavior and also show a simple case to reproduce it. I didn't know that Eric can't read a simple Scala code, but well, now I know.

And now Shridhar writes, saying, I have a case, it was working, and now it does not, what now? And Eric answers: it's probably a bug, I'll investigate it. And Prasanna pops in, giving his wise advices.

These people are busy doing stuff, and we just interfere, breaking things and distracting them from the important job they do.

As to Eric, I'm on the verge of stopping using his libraries, instead I'd rather copy-paste them to my project, tweak them, fix them, civilize them. Because I'm really tired of all this shit. By the way, I already had it once, at "Remote Control": I was on the frontend side, and I just wrote a monitor that was pinging me when something broke on the server side. I did not reach then the stage of "send an email to these idiots when their server stops working", since I was moved to code "bird view" for Android.

How can we teach programmers to write code? I don't know. There are books. But they don't read the books.

March 07, 2022

Oh, it's a work day today. We synchronized something with Stephen, like, he would help me with that shit that I'm bugging Eric, but I told him, not right now. Maybe later.

Eric is right, in a certain sense: if he can't run tests here's a sample curl command.

For that I had copy-pasted a method from his class (`AbstractBaseEndToEndTestBase`) to my "base", and added `val curl = s"""curl -d '$body' ..."""; println(curl)`, and that's it. Next time I'll need to add a creation of a Jira case right in the test. I'm tired of this, am I a QA or something?

But he had suddenly cooled down and admitted that he just can't run end-to-end tests, he does not have proper credentials. Wow, welcome to reality. No, I did not add credentials for him. Because why. I'll do it tomorrow, if he asks for them.

Then we had a standup, but I could not stand at that moment, because I was in a car, we were driving from a bank, and besides, we stopped by a "Southern Restaurant", where we had shrimps with grits (they also add sausage, and pour an egg on top... it's Carolina!).

So I was only giving my voice, and explaining for the fifth time, que pasa conmigo. As Vitaly Gordon says (we dislike each other, but I recognize his wisdom and intelligence), "the company could not invent anything better than getting the people together and giving them their daily tasks".

Eric was writing on Slack, from time to time, asking me when I'll give him the curl. And I was answering that It'll be ready soon. Then I posted my curl.

Then I spent some time improving my tests, so that they would explain, in simple words, what's broken. Also, found another bug of his (no, I wasn't looking into his code, it's just total nonsense, but here's a test, and it demonstrates that his code fails, what else do we need). So I created a Jira case for him, sent it to him and to Arie. Let him explain all that to the management. (Wasn't I working as a QA for him?!)

Oh, and we had another meeting after that, but I was late for it, I have things to do at home, not just working at work from 7am to 8pm. As usual, the Great Leader Prasanna was doing his show of Great Leadership, comrade Kim's style.

I don't trust this company, no, I don't. But you never know. I didn't trust Upstart, and see where they are now.

That was it for today. Sent a pull request to Stephen, adding that there's no rush, you can review it tomorrow. He answered that he's busy too, he went to his firefighters training, so he does not have time for that either. Wow, this immigrant is a firefighter, I'm really impressed. It would have never occurred to me... and yes, I'm jealous.

March 08, 2022

No fun.

First, Eric fixed it, then I fixed it, then I fixed it again, and it was all like this.

In short, I have to find a way to experiment and fix it without scaring the passengers.

So, I'm keeping quiet right now, just closed a couple of small Jira cases.

Tomorrow we'll talk with our PM. Stephen already went to talk with Prasanna and Eric, but what's the point. What's curious, eh didn't invite me to that discussion. Not that I care. But I plan to talk. to our PM, how can we schedule our work without these fucking surprises. We need to release, after all.

Stephen and Prasanna and Eric just agreed to kick out one subrecord, because "something is not working". Good job!

March 09, 2022

So far, my stuff, all that stuff, does not raise any complaints, although some end-to-end tests are failing. But now Stephen is trying to fix his part, and since Eric is helping him, everything's falling apart. Eric either doesn't know how to run end-to-end test, or doesn't want to. Today Arie wasn't present at the meeting, and everybody was arguing regarding "how come everything's broken here, and why

can't we run the server locally". Why, everybody knows why. Seb and Eric had recorded a video, two months ago, "how I run the server locally for the last four years"; they even convinced me to watch it at least once. Now I watch it carefully, making notes, and asking them: "how EXACTLY should I run Docker, and how can I install Postgres there?". Of course I know how to install Postgres in Docker, but you must have some kind of config file, do you have it? And how is it used? Where should I install it? Tell me (and I'll document it). In short, it does not work for anybody except these two guys, but they don't know how to run tests.

Shridhar, instead of scheduling a meeting with me, decided to bug me at the standup: "how come it doesn't work again?". Well, guys, I'm not tired yet to explain that Eric has blocked this subrecord, because his code can't handle it, and the bug is filed about interpreting this subrecord. Why can't I run it locally? See above.

Actually, I think I can, but I need a couple of days when nobody would be bugging me, and I'll finish it, it's not that hard, but it's bulky, and I need time without anybody distracting me. Meanwhile Stephen has a problem (created yesterday by Eric), that all top-level entities must have an id. Why t.f. do they? I understand, these are ancient beliefs of dumb devops, who were told, at some online courses, that databases must have an id for every record.

Should I add that there are no foreign keys in our database? Because Prasanna said that joins are not needed: they are bad for performance. You must have seen the logs, where their library queries the database for each entity separately, totalling about 10 times per request, give me this, give me that. But no join! These are separate queries.

Don't worry, don't worry. There are more serious problems in our lives. But, on the other hand, Bill has shown how he grabs logs from the server, via `tail -f` and he views right in IntelliJ all those exceptions in the logs. I'll have to copy Bill's scripts, I love them.

March 10, 2022

It's no fun, but my communications with white coworkers works much better than with the Chinese, can't explain. Just that this mutual respect and easy mutual understanding.

Can't explain it. With my Chinese colleagues I have to, all the time, get into small details, and the colleague seems to be insulted.

I ask Eric and Seb, in your video, you say, at minute 17, "launch postgres in docker" - how exactly can I do it? The answer is: "read the script". I have to read a shell script. Okay, I read it, and I see nonsense mixed with bullshit. In particular, it doesn't work for me. I fix it, run it, get to login, and it suggests to choose one role out of zero roles. I write to them again, asking. Fuck, can't we just cooperate?

OTOH, with Stephen, we can perfectly synchronize our work over the same module: he catches some bugs, I catch other bugs, and we coordinate. More, this morning these two Chinese geniuses broke

the database, and we were sitting waiting when they'll fix it. And everybody is guilty, except, of course these guys.

March 11, 2022

It's all stupid and crazy, but nothing interesting.

Stephen has found a hack, how to bypass that Eric's new shit. He intercepts JSON in the property and calls the deserializer. It works.

And the quest continues, "how can we run a service locally". Arie made me especially happy, writing on Slack: "Bill can run unit tests, Stephen can run unit tests, but Juan, for some reason, can't run unit tests". I wrote in response that unit tests are what I run all the time; Stephen added that it was who wrote MockService, due to which many problems with unit tests were resolved. I had also posted a quote from Wikipedia, "wtf are unit tests".

The rest was just nonsense and meaningless bullshit. Nothing interesting.

Suppose they decide that I'm too arrogant and fire me. So what. Are not there jobs out there? I can hang out without any job actually, I have my book to finish.

March 15, 2022

So, what happened today.

I had redirected my pr to Bill, as suggested by the previous reviewer. Bill wrote to me, on slack, some vague considerations regarding this solution not being very sane. I agree with him, and eventually wrote to him that I have a good sound solution in mind, so let's wait until I slap it together. He suggested first to share my ideas with Ananth and Eric. Eric is a nobody to me, regarding software dev, but I'll share. I'll slap together the code first, and write a description in the PR.

Then I took a couple of jira cases; one was easy, I implemented it and sent it to Jordan for review. But by the time of standup it turned out this thing is not needed at all; Sridhar just closed the case. Actually Arie closed the case, and I was pointing him where to choose on jira page the kind of closing. Strange. I guess Arie has more important things on mind these days: he's getting thinner and thinner, and spends tons of time on medical absence.

Another pr, I worked on it till about 20:30. A bunch of fields had to be added to an entity. I added them. Then I found that this, and a bunch of other similar entities, don't need to have an id field redefined, duplicating it from PiEntity trait. So I kicked it out. Tests pass, pushed the files, then decided to improve more. Define a common implementation of equals and hashCode

Now, after three hours of refactorings, a bunch of tests started failing. Stashed it, pulled what's in github, ran all tests, and sent it for review.

Around 12, I was busy, so I did not see anything, but it turned out that the current “Eric’s solution” does not work with my old solution, and Prasanna started asking, why I was not using Eric’s solution. I politely explained that it was Eric who used my solution but made it more complicated than necessary, and it does not work (I think).

In another thread, at the standup, they started asking me again about why did not I fix that subfield that still, assumingly, does not preserve the case of keys in a map. And again I explained to them that if we include this substructure, UI people will start screaming, because the records will just disappear (crash in the db layer). And the case is blocked. It’s blocked by the case that I created for Eric. Eric just closed that case of his, this morning, with 30 pullrequests merged without ever asking us for a review, and with one PR still remaining unmerged. But case closed.

So he probably felt guilty and started asking me, “what is not working in your code”. I slowly and in simple words explained that the thing does not work in his code.

And then he confessed. Suddenly. That if the data structure that their code (just a regular class referred from another class) does not accept, their code does not tell anybody at init time. They complain about the data structure when you send them that data.

Ok. This I can fix. Their “init” method. So I thanked him, and promised to make it work.

Meanwhile, suddenly, Eric just surrendered. His “very smart annotations” don’t work when they are inside substructures. So Stephen (not Eric, he’s “too busy”) rolled back all those shitty annotations (there was no used in them anyway, except that the database records look the way Prasanna wanted them to look), Stephen rolled the structure back to the way they were when I kicked in my simple solution. Cool!

Tomorrow I’ll go ahead and try to fix the two remaining end-to-end tests. Oh, also Arie managed to push Eric to answer, why t.f. they don’t run end-to-end tests in Jenkins, and show us how to set up Jenkins to run them. He showed us where it is, but did not manage to just make it work, enable them there; he may not know actually how to do it, and what to run. I’ll do it a little bit later, I still have two cases to finish.

Also, Santhosh, our brilliant QA, wrote a monitoring service that basically runs half of our end-to-end tests, once every hour - but it’s in his script.

Magic happens. Is it just in the air?

In short, about 12 hours of work again. I need a break.

March 16, 2022

I’ll be short today (okay, an 11-hours work day, with short breaks for taking food).

In short, that shit that Eric slapped together, I'm studying it, what's going on there. Added "saving the results of initialization", added logging. I got a feeling that no `init` is ever called. Added a method for pulling out results via http requests, to see them IRL. Debugging via HTTP POST.

Also the field that never initializes, I've split it into pieces, and started adding it into the "production code" (it was only in the test versions of the requests). Nothing is written to the database. Hello there! That's how `init` doesn't do anything. It's never called. Wow. Ok, I can do it some other way, we will see.

And another idea about flexible formatting of data filters on the "core pages". I'll have to write it carefully, and send the updates to those arrogant idiots who believe in their superiority. In our profession every idiot views himself (rarely herself) a guru and talks condescendingly. I had discovered it at Borland, an it amazed me then.

In the evening we had a meeting, with Arie and Jordan, discussing that "synchronization flag", a boolean field in one record in one table, which is supposed to mark whether the data are "clean" or "dirty". It seems like Prasanna changed his mind regarding Designing Concurrency, and delegated it to Arie to figure it out.

So, we discussed it; it seems like I managed to push a solution with an atomic pointer. There's no need to synchronize. We put the data into one heap, and get the data from another. As soon as we are done, we flip the pointer. This was called "screen buffering" or something, in the '90s. I use this trick from time to time, in a variety of situations. Another trick comes from FTN, which is now called "blockchain" (and has signatures, which makes it different from FTN).

Oh, whatever. If they don't interfere with my work, I just sit and work, not wasting my time on anything else. But... but, the nerves. I have to take nitroglycerin. Ukraine. It's an absolute horror. I'm not a big optimist, but, seriously, I didn't expect THIS from the Russians. Fucking liberators. Bombing a theater. Bombing a maternity hospital. I hate them. I hate them.

March 17, 2022

So, in the morning I wrote to Eric that his `init` method is never called. In response, he pointed at a log where `init`'s output is clearly visible. Oops.

That was a mystery to me. So I decided to intercept its actions in more variables (I also don't believe this ancient Java trick of "double checking" for synchronization - it was popular, then it was discovered that it does not work, then JVM was upgraded so that this thing worked, but it was, excuse me, 2008. Now there are more sensible tricks, and in Scala, it's just lazy val that does all this, afaik).

Anyway, later in the day Stephen told me that `init` is called in one instance, which is later killed, and the service runs in another. Fucking nice. Meanwhile, I found an http api to call that `init`, and called

it, and saw the Error Message! (That was the only place where you could figure out what's up with all this.)

Stephen shaved his head during the standup meeting. It was his pledge or something, for his birthday, and he was gathering money to donate to cancer research for a children hospital. I've no clue, St.Jude, or. I don't care right now. There are bigger things to care about, right now.

Ok, so at standup they started all this again. Asking when this "upper/lower case" will be fixed. I told them again I'm blocked, because the database is not being updated. Eventually Arie got Eric to the meeting.

And Eric explained the shit.

We can update our class structure, and then, when the service launches, Eric's code checks whether the db table can be updated (altered); if it can, it's updated, if not, they just drop the ball. And then the server runs and crashes on all calls using the new data structure. That's it. No "predeploy", no checks, no crashes when db does not match the code format.

Ok. It's doable, but it's not our team's business. I just wrote to "server people" that they better do predeploy, and roll back if it does not work.

Meanwhile, the idea came to my head. Why the fuck should I insist on having the new data structure in the same column that had an old structure? I'll give it another column name, and that would be it. But the field name will remain the same, so the frontend guys will have the same API.

So I told them, it'll be done by the end of the day.

But before that, our PM suddenly showed his "I'm raja" face. "Why do we have all this? Why is it happening? Why does it drag for so long?" Since to me a raja is just a shitty crybaby, I asked him, how come there were zero tests? And how come our infrastructure does not allow serious local testing? It's been six years, and the whole culture is still as if a couple of students slapped together a prototype. Compare it to FB or GOOG stories. (That last part I actually did not say, it's useless and insulting.)

Ok, anyway, I made a promise. Bill also shared the news that he now *can* run the service locally. Could say it earlier, but anyway. He invite me and Stephen to a show he can do right after this meeting (dragging already for an hour and a half). I said sorry, I have an urgent job. Stephen said, sorry, I have a bug in production to fix. Okay, we decided to do it tomorrow. Tomorrow is the day of meetings, about 6 hours of meetings. When will we listen to Bill's show, I don't know. Maybe in our morning, when California sleeps.

Then I started doing that trick, first as an experiment, with a small test structure (that's the same server that frontend people are using, so one has to be careful) - it worked, and I started preparing the big shift.

Fixed it all, sent to Stephen, he approved, I merged, launched build and deployment, and went for a walk.

Meanwhile, Arie was bugging me with slack messages, asking, so, so, when will it be done? And Jordan was asking me, what's the jira case number? Sorry, there's no jira case. We have a PM, we can ask him, where's the jira case for the problem I've been working on for a month.

Ok, ok, not time for worrying. The stuff works. Notified them.

Stephen has similar problems with some fields in his records. So I told him, here's the trick. Etc.

I need a break.

March 18, 2022

In the morning I continued cleaning the code, kicking out the test versions of entities, etc. Sent it to Stephen for review, he approved it, I merged it and deployed it. Tests are not there yet.

Also, Bill showed to us, me and Stephen, how to do an “individual deployment”: there's a script, but it requires tricky env vars, etc. All this must be properly civilized. We hadn't finished talking when the regular Friday series of meetings started.

While they were philosophizing and giving speeches, I continued working, for a while. When they asked me, how's my case doing, I told them that all the preconditions are dealt with, and I'll finish it. By the end of the day today. Okay.

And yes, I have finished it today, the tests pass, and the last case is closed (as well as all my old cases are closed), and only the new ones remain. Cool.

Also, Prasanna wrote to us, me and Stephen, that he's moving us to the “core team”, because otherwise we are “siloed” in Arie's team. Actually, we're supposed to help Arie's team, but in general will be solving general issues. Civilize our shit, step by step. I have some ideas. Like, there's no need to deploy to just check the compatibility. Also, we don't need a predeployment. Just send a byte code to the server, and let it check the compatibility. It's a known experience, we did it in Spark, we did it at H2O; the important thing is to be able to do it only in dev configuration, but never in production. That's it for today.

Oh, and they are still designing atomicity, locks, etc. Prasanna, at one of the meetings, had explained to Jordan how all this should be done. Jordan tried to tell him that we already discussed it Wednesday, but Prasanna does not get other people's ideas. So now we have two keys in the database. Then they were interrogating me, how do I implement transactions, locks, race conditions evasion. I told them I have not invented it yet, but I'll investigate. No big problem, eh. They need atomicity to change a field inside JSON in a database. Will do.

March 21, 2022

In the morning, frankly, I didn't do anything, except synchronizing with Bill and Stephen, when exactly will we sit together and continue our investigation.

At 1pm we had a standup; I confirmed again that I've finished this thing that I was working on for three last weeks; finished it successfully, the tests confirm it, etc. Then there was again all frontend stuff; I was yawning and opened Turbotax on another computer, to do all that, the sooner the better. We are going to have a crazy amount of money to pay this year.

At 2:15pm we had the next meeting, a "big standup", where Prasanna gets all the engineers, from all the teams, together, and goes into the details of the whole product, what, where, when. Various people were telling various funny stories, using smart words and beautiful technical language. Then it came to our team. Prasanna solemnly asks, who's going to report? (Our PM is sick, got Covid from his kid, that's what kids always do, spread diseases around the world.) Arie pops up, and describes everything, with a lot of positive attitude. How Vlad has already finished implementing "custom filtration" (that is, a predicate depends on the record type - I do have this pull request, but I want to make it more civilized, I hate the ad-hoc code); and how everything's fine. I'm now some kind of shock worker, at least I'm viewed as such. Stephen is a production man, he deploys to production, a reliable guy. And Bill is a philosopher, a wizard; his code is not always perfect, but he's still pretty advanced. Well, with his age... (he's 10 years younger than me, actually).

After the meeting, I did some stuff on my home, and at 4:30pm we got together, Bill, Stephen, me, and we were running this "personal aws deployment". The code, the same code, worked differently on my machine, compared to Bill's. I couldn't get through SSL, and for Bill, the server behaved when talking to Postgres. Bill helped me, and I tried to help him with Postgres, but then we postponed it till tomorrow. Stephen was experimenting on his machine, but he wasn't disclosing. Or, maybe, he was just watching us, how we do.

I don't think it's all serious at all. An AWS deployment, from my village, takes over 20 minutes. I can as well merge it to master, build it there and deploy it. But it makes no sense; deployment is better done into Docker. Except that our dimwits have security all around. Why anybody needs security in a test environment? If they could stop interfering with what we do. There are layers of code development, you can't write a "hello world" and deploy it to AWS under all that security.

After that meeting, around 6pm, I drove to Lowe's, bought a bag of soil for my trees, and a long hose. Now I can connect two hoses and water the pink decorative apple tree that I have planted by the wood, along the driveway.

In short, I was slacking. Ok, I was tired. Ok, I'll work now, for about an hour.

March 22, 2022

Today I didn't work much, mostly I was just thinking.

At the standup Stephen started whining that, see, we have one data structure, but in the code we have another, and it's incompatible. They started buzzing about all this, with not a second of a break. Eventually a break happened, and I suggested Stephen to get together after the meeting, I know a solution that's relatively universal (I already wrote to him about this, but he didn't seem to get it). Well, okay.

After the meeting, when we were scheduling a Zoom meeting, he suddenly writes: oh, I found a bypass. Well, okay, do I mind? Stephen likes to ignore his advisors and do it the way he feels good about. An Irish man! BTW, he has his birthday on St.Patrick's day.

Then I refreshed one pull request, and sent it for a review; and asked David and Chris, on Slack, what do you think, guys? And here Shridhar pops up and writes that hey, we've done it, we wrote a filter, no need to change the database.

Well, nice, I wasn't planning to change the database, but if the issue is closed, that's it, I'm not volunteering anymore.

Meanwhile Arie writes that there's another urgent case. We need to atomically update a field inside JSON in the database.

I started digging deep into all those crazy methods in the "database service": there are zero tests, but lots of murky code. Then I asked on Slack: guys, can I do a transaction, or, say, can I do a CAS. Now David writes: what's the problem, it's just "update something set a = b where something.a = " etc. Oh shit, he's right. But our "update" does not have a filter, so we can't write a "where". I'll take another look. As a last resort, I can just send a SQL statement.

This is when I ended my work day. Mostly spent in thinking about solutions, while planting my trees. But now I'm done with thinking, and tomorrow I'll be implementing these things, this or that way, we'll see.

March 23, 2022

As to transactions etc: I continued interrogating them, until Eric told me that they have global locks there. It remains to start using them. You can do it via a service, or directly. Well, ok. I wrote an "atomic update", and implemented some transactionality. Next I'll have to implement CAS. It does not look like a big deal.

That other thing that I wrote, about "custom filters" < I can't get a review from anybody whom I asked. I write to them twice a day. Silence. Wrote to Arie. Silence. I'll have to bug them individually.

What else? Nothing, basically. At the standup Shridhar started interrogating me, what's up there. I ask him: are there any docs? Sure, he says, there are chats on Slack. meaning, go figure yourself. But how about the docs? And now Jordan sends me a link to a doc, the one that he promised to write, last week, after our discussion. Thank you! I took a look. It's well written, but it won't work. Why? Because Prasanna, when Jordan told him on what we agreed, me and Jordan, answered that it's nonsense, replace your bit flag with a timestamp. Ok, sure, I can, and I'll add atomicity to this shit. But it won't work. But if I tell them, nobody would even react, because Prasanna Said. Well, fuck them all, I don't have time for all this. I have my taxes to work on, it's complicated this year.

March 24, 2022

Again, nothing special. Just busy working, looking for a way to implement CAS (no big deal, actually), and how to do partial updates (here we have problems with JSON libraries).

March 25, 2022

So, I discovered this: if you have `Option[scala.Long]`, Java reflection tells you that the “generic param” has a type `java.lang.Object`; but if you have `Option[java.lang.Long]`, it knows that the “generic param” has a type `java.lang.Long`.

Which basically prevents correct transformation to Json. Funny, no?

Anyway, when it was discovered, I fixed my shit, and the tests passed, etc.

Ok, but actually we had tons of meetings, as usual for a Friday.

Ok, and when the meetings ended (and I promised to do something soon), in the evening, Eric(!) wrote to me that there's a new Jira case that Arie created, that needs to be done by Monday. Ok then, an emergency. I took a look. It's a pretty dumb change in the data structure. Can do, though, if there's a rush, right? It's half past nine, and I'm finishing it now. Won't open my work comp during weekend, though.

A recruiter called me today. I told him I'm fine where I am, and he said he has a bunch of good Scala professionals, and I promised to talk to Arie regarding this.

March 28, 2022

About this time, almost 11 at night, I hardly remember what was in the morning.

Ok, Eric had approved and merged my PR probably on Sunday, so I just deployed it to dev, and ran the tests. Everything's fine - but the frontend guys got puzzled by the sudden big API changes, so I had announced clearly what the hell was going on. Ok.

Then there were one meeting and another. I promised to do the first of the two remaining steps today - but I could not. It turned out to be trickier. Only by the end of the day I discovered how Java reflection is a little bit crazy (I should stop using Java reflection, I guess); so I found a way to transform an object into a map of named properties (records or simple types), but did not implement it yet.

So that was the day. I have some tests, they don't pass yet.

The general trick is that ok, we have a record, and we have a delta, both are trees, and I have to carefully merge the delta into an existing record tree, keeping in mind that the structure is all visible to the frontend guys, so it should not be a surprise - and it also should be strong-typed.

Anyway, working on it. Tomorrow.

Something happened to Ben. He's still in slack, but he is not seen anywhere, and everybody said goodbye to him last week; I asked, on slack, what's up, but there was no answer. Weird. If only I cared.

March 29, 2022

Nothing interesting today. Ok, I've merged my pull request, but it a) broke the build (how come it compiles for me, but when peacefully merged into master, it does not?) b) I broke four end-to-end tests. This second case is no biggie, I know where NPE is coming from there, so I fixed it, and besides, finished implementing what I wanted to implement.

Bill approved it later in the evening; I'll merge it tomorrow. Will have to watch it, since we don't have a precompile with a rollback on failures. I'll have to figure it out. Now I don't have any Jira cases, except for playing with the build system.

That's it for today. As I said, nothing interesting.

March 30, 2022

I was busy all day.

First, I slapped together this new API, reported the fact on Slack, then decide to write end-to-end tests, after deployment. And then I discovered that the API per se does not exist actually. Well-well-well, then I sat down and spent time until 9:47PM implementing it. Including the tests. There was a tricky trick with an atomic update of a field inside a substructure.

Not that important, actually.

There was another strange thing. I discovered that I don't have access to Jenkins anymore. I wrote to our devops about it. They sent me a link to a Slack channel called "All Engineers". Never heard of it

before. The trick was, I had to remove several lines from etc/hosts. Meaning, these guys are cool, they decided to better organize the DNS, wow.

Meanwhile, github suddenly tells me to authorize via Okta. Very funny. Will do it later. I don't even have Okta installed yet. WTF, really.

Anyway, no biggy. Isn't it convenient, to do in parallel house chores and the work stuff. Like I did in 1992.

Hardly any communications with the work. Just a couple of words with Bill and Stephen. Arie is busy, he's taking his kid to school every morning, he does not have time for work.

March 31, 2022

This morning, I was horrified to discover that I'd slapped together everything for that new functionality except for an actual web API. Oops. I quickly wrote it up, ran the tests, sent it to Stephen, he approved it, I merged it, and deployed it. I wrote an end-to-end test for this thing, but the tests were behaving strangely. They depend on the order of run. Also, I was horrified to discover that time was freezing in one of my simulated clock tests, and the before function wasn't resetting the clock. Oops. A global variable. Well, I had to fix it. Then, at the standup, Arie casually asks what's going on with the case, and I kind of talk too much, like, the API is done, and I'm going to merge merging the end-to-end tests ASAP. Well, he says, it's kind of contradictory; either it's ready, or it's not ready. I say, "It's done, it's done," but it's impossible to simultaneously slap together the code before deployment, and have the end-to-end tests too. I had explained this several times, but he still doesn't seem to get it. So, in short, I say, "It's done." The task is complete, everybody's smiling. Then they spent a long time droning on about some front-end topics; to me, there are more important tasks, but their thinking is a bit shallow. In my opinion, the front-end people don't understand back-ends or functionality at all. They just want to see the forms. And we don't understand their shit. I can't say I've never done front-end work (see Google translate page). But we're looking at it from the p.o.v. of implementation complexity, while they're interested in all sorts of not so important stuff; and they don't see the whole picture. Oh well, it's all unimportant. Then I spent the evening experimenting with scripts for creating a "private server" in the cloud; the problem was that I have AT&T internet here in the village, which is fucking slow, and the deployment is done via scp (not via rsync), which is why everything's slow. I hope to fix it later, later, not right now. First, I need to make it work. And anyway, why bother sending huge jars? One can sync the sources via rsync and built it all there, in the cloud. How many source bytes are there to transfer? See, now it's sending a gigabyte for some reason. Okay, whatever. While it's sending gigabytes, I have time to do my own things. Filed my taxes, for example. A shitload of money, but considering that we sold the house, it's not even that much of a deal. If California didn't demand its cut, it would have been absolutely amazing. But as it is, give them 12k for selling the house. Not that bad, of course, considering everything. I've been messing around with taxes for the last week. I've

saved a ton of money, basically. If you do it right. I had to pay Turbotax about 350 bucks. I bought the most expensive package there to avoid the hassle later. It's fine. I consulted with real accountants twice. And now I'm trying to run tests on a freshly deployed server somewhere in the cloud.

April 2022

April 01, 2022

It's kind of funny.

I kind of started to figure out how to replace scp with rsync. It's not immediately trivial.

Stephen has moved my generic classes from our module to the common one. Good job, I like it. Well, I don't enjoy a 60-file pull request, but... But everything is extremely clean. So I approved it.

At the standup (we have meetings the whole day on Fridays) Shridhar started talking total nonsense, about a Jira case for me; I started philosophising about "how to correctly design API", and he's fixed it immediately! Profit.

In the end of the day Vignesh (a manager from the other team, where we, me and Stephen, were deported as being too smart and able to fight the Chinese mafia) wrote that he'll be glad to talk to me next week. I wrote to him that I'll also be glad to talk to him.

Well, it would have been nice to experiment more with rsync, but I don't feel inclined to do it today. Will work with it tomorrow. My taxes are filed, so I have time.

Don't I talk too much? Although, who fucking cares, right?

April 04, 2022

This morning I checked again how rsync works instead of scp: everything works ok, it was just a glitch.

Deployment works ok, and is much faster. But there are new troubles. Some secrets should be extracted from cookies in the browser, and copy-paste them in the script. In short, it's a nightmare. And it doesn't work. And you also have to copy the ip. And so on.

But then I had other things to do beyond work. Communicating with friends hiding from Russian rockets in a basement in Chernihiv threw me off the game. I could not seriously pay any attention at the meeting to those disgusting voices of our PMs, I could hardly think about the work, so I told them I'm not feeling good, and went to mow the lawn. This I can do.

In the evening I took one Jira case and fixed it (it was about injecting two lines in the right place and removing two unused methods).

It's building now (tests passed); then it will slap together a pull request (I have it automated), and I will only need to ask Stephen's approval for this pull request.

Stephen, by the way had sent me today a pull request for review, 43 files. But everything is technically correct there, everything is meaningful, and it would have been really hard to split it into smaller pieces. So, okay.

Wrote to Arie that I'm not in a good state to work these days, so I'll be doing just some easy routine. And maybe starting from April 14th will take a week of PTO.

We will see.

My nerves are running out. And what's waiting in the future to come? Why are we such snowflakes, damn it.

April 05, 2022

Starting in the morning, I was kind of slow. But managed to slap together a new pull request.

So far, everything quite.

But then production crashed (a little bit). One record (out of about 200) suddenly hung on read, for about 60 seconds, because Spark was hanging. How come Spark is involved in reading data from the database, it's an open mathematical question. But since it's production, there was lots of noise, and in the end it was discovered that the last one who calls anything on the server (some unrelated service) it's me, that is, it's my code. Yes, nobody shows me the logs, but Bill found out. Then Bill spent a lot of time analyzing the stack trace (not being familiar with that code)... but still, how is Spark related?!

And now Eric pops up, saying, hey, you are using a wrong service. How come, everybody uses that service (there are six use cases), as far as I remember. Wrong service! We have a new one, which is the right one, and that one that you used is wrong. Oh, nice! Okay, I say (we are all on Zoom), I'll fix it tomorrow. No-no, Prasanna ordered it to be done urgently. Okay, no problem, let it be urgent. And then Bill comes over, and started modifying the code! Hey, Bill, no! We already did it, modified the same code in parallel. Leave it to me!

So I started modifying it, switching to the new service. BTW, injected some generics (the new code is more advanced, it's new!) Now there's about three times less code.

Okay, I wrote it. But the tests don't pass, because the mocks are mocking the old version. Oh shit, whatever. And I decided: forget about these tests for now, I'll fix them tomorrow. It's already almost 11pm. Sent a pull request to Bill. And closed my work computer. If anybody's willing to merge and deploy to prod, you are welcome; I want to sleep.

And that old API, which is kind of too general (we don't have architects, we have a CEO instead, and we code whatever gets into his head), and that old API is exactly what is not working. It behaves like this: after retrieving one record from the database it goes ahead and starts retrieving some derivative data from Spark. Oh shit, can you see?

Whatever. I'll fix the tests tomorrow.

April 06, 2022

It was a rather stressful today. They have approved my yesterday's pull request (they did it last night), and I've merged it, and there's something wrong there. When running on dev, some of my tests pass, and some don't; and our QA found it too: we can't serialize `Nothing$`. And there's no stack trace.

So I had to update it, to check out the sstack trace.

The stack trace shows that, haha, I have a method, with a generic, and the generic has a type limitation, from below, and there's no upper limit for the type. As a result, if we skip the generic, and the result is not `T`, but `Option[T]`, then Scala assumes that `T=Nothing`.

The solution is simple: have a phantom method,taking `Nothing`, and if somebody calls it without a generic, and the parameter is not calculated automatically, we see two alternatives, so the code won't compile! QED.

So I implemented that fix in the evening, sent it to Stephen and Bill, but they are already asleep, and as to the people on the West Coast, I just notified them about this.

Okay, okay. Nobody told me, but it seems like Vignesh is my new manager, but they are not inviting me to their meetings, and I go to Arie's meetings. Arie is also absent, he either is sick or taking his kid to the kindergarten. I got a feeling he may have health problems, but he's hiding it, American style. Who knows. America.

There was an engineer at Borland, he was sick, but he went to work until almost the last moment, when they took him to a hospice. There's a stone in Borland garden, dedicated to that guy.

Bo works from Sweden, and, Swedish style, continues askin us, backend guys, when it will be ready. I ask him: Bo, if you start writing some code, do you know when it will be ready? Silene. It makes no sense to explain this to managers, they just don't get it. They count everything in "points", and they don't even know how to add points. (I wrote on LinkedIn that, to do it correctly, you have to add logarithms, but people are, on average, pretty dumb, so there's no reason to expect understanding.)

That's it for today.

I still have a bunch of tests turned off in this code; will fix it tomorrow. At least end-to-end test pass, mostly.

Oh, also, talked to David. He shared his opinion that to access other modules, one has to use REST HTTPS, not Java calls. I think I agree with him. In this case we won't need to stuff all the code into one huge jar, but we can cut it into neat chunks, and develop independently.

April 08, 2022

Well, not everything works ok after this small API revolution; two types have problems, so I started fixing them. Made a mistake while fixing one of them, I took the name of the record from another type. Oops. Good that Prasanna showed me the problem. Actually, we have to fix it, to make the compiler substitute the names. There is a way to retrieve the companion object, Miles Sabin, the communist, discovered the trick about 10 years ago. I'll do it. Later. Fixing the tests, step by step. By the end of the day all the cases were closed... seems like. Especially the one where Santosh was extracting a wrong field value; and I look, and see that it's my old test that injects this bad shit, into the standard record (shame on me); I've fixed the test, and ran it, to make sure everything's fine.

The people are, a little bit, getting crazy: where's this and that column? Then Jordan asks: "what's this column, Case Export Parameters MAC9951?" I'm trying to explain to them that during large changes of a structure field I rename the column, to make sure the old data are not interpreted incorrectly. Haha. The people got kind of angry, but then Prasanna showed up on Slack and started explaining the principles of data and API organization, that it's not a good idea to go to the database with raw SQL if you, guys have an API, etc, etc. Everybody, as usual, agreed with him immediately. Thank God, we have Prasanna to teach them.

April 11, 2022

An awkward day.

I started with writing a pull request where the database on the server updates not just one tenant, but also my private tenant (a tenant is just a namespace), as an experiment, since my tenant doesn't get updated. As a result everything broke on dev, and Eric a) fixed it fast, b) blocked the deployment. No, first he blocked the deployment, so Jordan was having problems with his piece, and he begged to fix it, but the deployment was blocked. I also had a small pull request... but that's nothing.

As a result, I started creating a task in Jira, requiring that our tests run. Late in the evening I finished it, and sent it for review.

Arie was present at the meeting; he looks good. Thank God, I really thought that he was seriously sic. Me and Stephen, we are having just our last days in this team; okay. Told them that I'll be on vacation, starting Thursday.

I still have a couple of tasks... or just one, the one I put in Jira, and the other one is also mine, but it's not in Jira. I should have put it in Jira, too. Well, whatever, who cares. I'm not doing much these days. Just attending the meetings. Signaling my presence.

Yes, after all those troubles of the last week, I need a break.

Oh, by the way, they have another P0 bug, this one has nothing to do with me. So I'm on a bench.

April 12, 2022

At last I made Jenkins run my task, with the tests for our module. Informed our team. Jordan was happy. I'm not going to make it required yet, I'll be running them manually, then will set it on a timer; and we will see.

At the standup, a new PM started wondering: "why do we need tests?" Wow. Me and Bill and Jordan started explaining to this idiot the meaning of good software development technologies. And I checked out the guy's profile on LinkedIn. Zero professional education or experience. An experience of bullshitting and bureaucracy. Of course, nobody cares about his opinion; if something happens, Prasanna will give him spanking. That's the advantage of having a competent CEO.

Also, it turned out, at the standup, that after updating the API some clients stopped working. I got a bit animated, and asked, how come we don't specify the version in the API, and don't support the old version, for a while? Now Jordan and Bill were happy to hear this question, and we agreed not to do revolutions anymore, but do development and deployment step by step. Less panic, more common sense.

Then I had no desire to work, so I went to my garden to do my stuff there. Next, ideally, I was supposed to play with a local server, but I was too lazy, so, instead, I started cleaning after Stephen's migration of my code into the common library. Step 1: remove code duplication.

That's it. I'm going on vacation tomorrow. Not exactly: we have some meeting, and a discussion of this P0 that I spent three days fixing (the local law is that it can only take two hours to fix any P0 bug - but I leave it to the PMs to solve all such problems in two hours).

April 13, 2022

It's all nonsense, frankly. I'm writing it in the airport, CLT.

This morning I upgraded tests on Jenkins, ok. I set them to run thrice a day. We will see.

There was a standup, and I was explaining to our PM again, how computers work. At times I almost feel like agreeing (no, I don't) with Timur, one asshole from Kiev, with whom, a while ago, we were sharing a desk, that some Indian programmers are a bit dumb. Some are. But not the female Indian programmers. These are the brightest ones! Just BTW. I guess, it's a natural selection; it's really hard for a female to get through the glass ceiling. The meeting dedicated to "what broke last week" was canceled: actually I had documented everything, so there were no questions. Eric added that we had already discussed it, at another meeting, with zero docs. All the information is in Slack comments or just side comments during the meeting. Oh, whatever. Canceled? Good. Eric can't make it today, I can't make it tomorrow.

Nest I made a stupid mistake: I merged my pull request the last minute before taking off, and the

minute after that I deployed it, and then I took off to the airport, and, while in Uber, I was getting messages about something not working. Stephen writes: “Prasanna gave an order to roll it back if it doesn’t work”, and I answer: “please roll it back”. But in reality it’s not clear whether it does not work. So I’m sitting in the airport, checking my tests. I have 30 minutes before boarding. All tests pass actually.

Still, I should not have done that.

April 15, 2022

People woke up at work, and they are writing to me (two Chinese guys and one Stephen) that my Jenkins job does not run any tests at all. I answer: how come it doesn’t, check out the logs, they even show a couple of test failures.

Silence in response.

Do I care? I’m on vacation.

April 21, 2022

“Ok, the heartbeat took 16 seconds to respond”

“dashboard has 50k charts associated with it”

“search method is implemented as a fetch of a list of all dashboards”

(from chats) Work, work, work. I planned to get up early, do something useful (like, fix my pull request that did, according to rumors, caused a crash... but there was no crash).

Ok, took a look. David suddenly decided to “improve” my latest updates: rename a class, insert a dot in one place (in Scala 2 a dot is often not required). All that stuff. Okay, okay, I don’t fucking care these days.

By the evening I kicked out this renamed class, having sent to Stephen a 22-file pull request (I have actually pulled out the remaining functionality into the class `PiData`, which is now for me something like a basis for all our usual classes (request, response, record)). There was some more stuff to do, but I have to figure out OpenAPI. At least add tests, to check whether anything changed after an API update.

Then I opened Jira, found that I have a new case, to clean the code in general in our codebase. I love this kind of stuff. But the case is not well-formulated, so I got questions.

At 1pm we had a “standup”, and they were buzzing and buzzing, and only at 2pm I managed to ask a meta question: “can I ask a question?” Anil had gracefully agreed (now they are all some kind of rajahs), and I asked: what exactly do we mean here by “Application”? Nobody knew, and they sent

me to Vignesh. Seems like Vignesh is my new manager. Arie is leaving. It's a pity. Where is he going? Another company? Or worse? Is he okay. It's not clear. I wished him all the best. Although there were cases when he messed up (who doesn't), it was really good to work with him. It's a rare case in the Bay Area.

Now we had a 1-1 with Vignesh, at 3pm. We were like two insects with antennas, touching each other, setting up a contact. Spent some time remembering our so-called alma-mater, WeSell, fucking WeSell. I did not say a bad word about the company. Vignesh listed managers in our team over there, a bunch of Indian names, I don't remember any. Well, that's me, I hardly remember those with whom I did work. All that management, I forgot them immediately. Do I need them? Fuck them. I have really good words for less than half a dozen people at WeSell. And there's one Ukrainian, a decent guy; we chat with him.

So I asked Vignesh, what does this case mean? (It was Vignesh who wrote this case.) Vignesh doesn't know either. So, I say, I'll write a doc and will be asking the people around.

Perfect. I immediately wrote the doc, and sent it to the guys on Slack. But the guys were confused. EspeciallyDavid: "why do we need it, what is it for?". Eric did not get involved, but then Prasanna showed up and explained it in plain English.

Okay now. Any comments to my document? None. And now what? I can remove all those calls, if nobody cares.

And I started. But then, suddenly, Vignesh went ahead and changed the whole content of that Jira case. No I have no clue what they mean and what they want. Of course I have something to do (we already discussed it with Vignesh, how we are all busy at work (no kidding, we spent 3 or even 4 hours a day working), and we will work harder and better.)

No, I understand, Vignesh got scared, and started changing the case after the stormy discussions on slack. But now this "description" (the title of the case is the only description) makes no sense, and it's just general words. Like slogans in the USSR, "workers guarantee to the five years of quality").

But as a result, I feel like I'm free to close my work computer.

Do you think I'm complaining? No way. I'd rather complain about the stocks that I wasn't successful with. But there are successful ones, like TSLA. And maybe NFLX. ï

April 22, 2022

Nothing important, in general. Spent time investigating, how come, after running the tests there's no file with test results. So far it's not clear. My suggestion to Eric and David, that I pull out from reportingService the functionality that nobody, except such and such service, is using, was not well received.

Eric, in his style, writes “don’t touch it if it works”; I write to him: okay, if it works for you for the last five years, without problems, I won’t touch that part. David writes to me something instructive; he considers himself the smartest. I remember he was editing my code, replacing spaces with dots.

In short, that’s what I needed: confirmations from these people that nothing can be done.

That’s it. Also, Stephen sent me a pull request, I approved it.

All that stuff. Then Alex was convincing everyone (Friday is a meetings day) to get security training today. That’s what I’m doing. In parallel, I’m moving the content from my old phone to the new one; the process is now very smooth. The two phones sniffed each other, negotiated, and agreed.

Meanwhile, on my work computer they are teaching me to never open files received by email.

Oh, we also had a “planning meeting”. We were discussing what we should stuff into “sentinel” data and API. I think I managed to convince them that this should be flexible and modifiable, and I’ll do it, and there’s no need to create tons of docs, we’ll just do it. Jordan is saying the same thing. So we agreed on it. You are probably bored reading all this. To me it’s like a status report.

April 25, 2022

In the morning I went for a walk, and, during the walk, I figured out how to catch this problem that testing logs don’t go anywhere, while Eric’s logs are okay. Added, in Jenkins script, this line: `find '**/*result.xml'`, and started the build. And went shopping.

When I returned, I found out that this build was stopped. I ask, on Slack, who stopped it and what’s up. And now the shit hit the fan. Eric writes: you can’t run end-to-end tests in parallel (he has a test, I have a test), because... because... because, as I understand, that this deploys (ok, he slapped together about two millions lines of code during these six years) is not so big in programming. Seb also popped up, writing, how dared I to make changes in a script without asking for a permission. This script runs three times a day, what’s the problem, what are his objections?

So I ask him, what’s the side effect? Something started working worse, - answers Seb. What exactly will be bad from running `find '**/*result.xml'` three times a day? Silence. He didn’t bother to answer. Wow, what the fuck. Two crazy idiots. Eric is an architect there. Fine! If I were younger, I’d have got out right away. But at my age, I just don’t care.

Nice, nice.

Also, I was cleaning the code, all these classes like `PiData`, `PiEntity`, and their derivatives, which Stephen had stuffed into the “common module”. I even have a Jira case for that.

Later we had a “standup”, and two more Jira cases were found for me; a simple one and a conceptual one. Okay, will do it tomorrow. That’s it for today.

What if I ask Prasanna whether Eric is going to cooperate with colleagues or he'll continue talking down, muttering orders through clenched teeth> But I know what Prasanna would answer. That we have to respect the hierarchy. Well, I kind of respect the hierarchy. If John Baez, or Martin Odersky, or Dan Piponi, or a bunch of other respected people write something, I read it with full attention. But these farmers Johns, this rural hierarchy, I just don't care. I know, some people are creative; but if they learned something serious (e.g., Scala), that would have been great. But they didn't.

Why should I bother? I'm a nobody for them. They are nobody for me. Why waste time on all that nonsense? I don't want to waste time on all that.

April 26, 2022

Started the morning by “improving” the code, then dropped it, having decided to get from the founders, why t.f. we are using Spark serializer, which is known not to work with Options. Kind of agreed, with David and Eric, that we should migrate to the “normal” one, I mean, serializer. And then I'll go ahead and simplify it. Ok...

Colleagues wanted me to do something like “delete this thing from this service”, that is, that I write such an API. I took a look: NP, I already have this method. So, I updated the Jira case - and kaboom, Prasanna pops up, saying, it's not what we need! There are two fields, with “the results of machine learning”, and we need to give users the opportunity to clean these fields, atomically.

Wow, ok, no problem. I already have the atomicity; all that was needed was to write a couple of entry points; so I did, slapped them together and sent for review, to Stephen, Bill and Prasanna. So that Prasanna would share with me his immense wisdom, how this API should look like.

That's it for today. Also, Anil wants me to introduce, urgently, a new entity. Sure, with pleasure. But this entity is supposed to be widely used, so I asked around, where to we stuff commonly-used entities. Seems like we found the right service. Good. I've updated the case in Jira; will write the code tomorrow. No big deal, again.

Other things worry me more these days; work is on the side. It's the war.

April 27, 2022

In the morning I noticed a message from Santosh, on Slack - oops, such and such business unit is not visible on the dev machine. Ok, started investigating it.

By the end of the day found my own bug. We have about 150 business units (all are test ones, except two), and by default, our style, we choose a hundred of them, then we filter them. Which 100 are selected, depends not on the phase of the moon, but on the current picosecond on the load balancer. Ok, fixed it now.

What else? I'm introducing a new entity. Besides, fighting Jenkins, in my free time. Also, I'm trying to fix the serialization (use the right library and don't use the wrong one) - but these fucking idiots in "core team" don't seem to be interested to support the right one. Why would they? Seb even didn't like it that I've just fixed their failing test.

Oh, fuck them. I don't have a mood for work right now.

April 28, 2022

Can't explain it, but it's 9pm, I'm feeling a bit dizzy, so just main points. And yes, I'm still running end2end tests.

What was it today. Turned out that my trick with setting limits for listing some shit does not help, because UI people send their on limit. Ok, ok, I'll have to fix the server code. Did it. Santhosh wrote (while I was walking in the neighborhood) that everything's fine now. Ok, but not for my tests. So I'll have to investigate tomorrow.

Another piece, Jordan found bad behavior. And I found that tests for that code were disabled by me, on 4/11. And since then some code got rotten. Oops, me again.

Also, can't convince the Chinese to approve my pr with a better serializer. Shit. Whatever.

Krzysztof told me that he got out of Video not paying them back the hiring bonus, after consulting with a lawyer. The trick: blame them of racism. What t.f. racism? Chinese one. The Chinese were maltreating the white people (of whom he was the last one left). Now he's in FB.

What else? Well, yes, 1-1 with Vignesh, and he suggested I write down my plans and actions, in Jira. Sounds pretty reasonable to me. Will do, but not today. Not tomorrow. Maybe tomorrow. Just have some urgent stuff to fix, and some code to write (that one does not break anything).

That's it for today. I'm going to sleep.

April 29, 2022

Another day.

This morning I've decided to work, for some reason. In short, two problems.

1. If, during the server deployment, the database wasn't upgraded for some table, then a) read the logs, b) other tables are not upgraded too - but nobody reports it. Fucking idiots.
2. When I try to list business units, and ask for a hundred of them, I get zero units. But when I request 101 units, or 99, everything works fine. The question is: WTF? Investigating it. So far Eric is the obvious culprit.

Also, my pull request, where I change the serializer. Wrote to David personally about it. a) Am I just being ignored, is it nice? b) Any problems with this pull request?

David (who isn't Chinese) wrote to me that application developers are not entitled to make changes in core code (fucking idiot, eh), b) if we dare to change it, nobody knows how spark will behave on our data (fucking idiot, eh).

So I answer: "Okay, I understand."

My understanding is that if I don't touch their shit, but I can write our own library, big deal.

After which David wrote to me that it's Spark that should be fixed. Well, I agree here, Spark should be fixed, but my experience with Databricks was absolutely negative (up to the troubles communicating with their CEO), so fuck them.

So, we too, David and me, like two bugs on a lawn, or two animals in the woods, sniffed each other, peed at the nearest tree, and decided that we should not have anything to do with each other. I'm fine in "applications", nobody interferes in what I'm doing.

I had a similar shit at Borland, a bunch of idiots from JBuilder team... oh, fuck them all.

I dislike it so much.

Then we had a bunch of meetings. Our CEO was explaining that it's probably time to raise the level of application quality from POC to production. (As if anybody here knows what he's talking about.)

In the end, at the standup, we agreed with Bill that we have some shared functionality, and we will join our efforts. Then I was going to work, but I got, on LinkedIn, a message from Ilya Eckstein that Ilya had died in October. Heart. At work (at Google).

Fuck... he was so young. Just over 40. How can this be! He has a little kid, about five year old. How's that. How horrible.

In short, I can't work today. Just can't. How come people die like this. So sad.

And now Facebook. Joe Nuxoll posted a pic of him with Cal Quinn. It was Carl's last picture. Carl died from Covid two years ago. Oh. It was Calr who let me fix protocol buffers at Google. It was epic. For me. Carl did not even notice it, for him it was a small thing, he forgot it fast. But for me it was epic.

So sad. So sad.

May 2022

May 02, 2022

In the morning they sent me a form, where I'm supposed to sign what I'm supposed to follow.

Including something about blogs, that I can't write anything bad about the company. Which company? Where Juan Gandhi is working? Okay, I won't.

And the good part was that I was gradually investigating the causes, how come I have this column, but it's not in the database; how come, the schema is supposed to get updated, right?

At the big meeting (it's Monday), when Prasanna asked how I'm doing, I said that everything's fine, but we can't modify the database without transactions: how can we roll it back if something breaks?

Now Seb pops up, saying "you have a bug in the code, that's how migration does not work, you must fix your bug, then migration will work". Well, I'm not at fault here, my schema is okay, it's another schema change does not pass, and one can trace the causes only with the police, in the log. But don't we need Atomicity? I was explaining all this to him loudly and slowly, like a teacher of computer sciences.

Now another Chinese guy, Eric, pops up: "we have all that, we are giving you a lock" And how are we supposed to roll the change back? - ask I.

Now Prasanna pops up and starts teaching our architects (two Chinese friends). He's teaching them the elements of ACID. These two immediately start nodding and agreeing, saying, yes, this is an important thing that should be done immediately, we need it, we need transactions.

That's how it works. In the old days I would swear, go talk to the management, etc. Now I go for a walk and jump into my pool. This is not my business, but I'm always glad to help.

By the end of the day I slapped together one pull request, while another was waiting for Bill's review (we agreed that he'll look at it, I'm extracting a trait common for his and mine code).

And I also got it how come the schema modification is not working. I did implement this shit with representing complex data in the database some time before Eric copy-pasted my solution and added some kind of annotations. Now his solution does not work for the existing data structure that I wrote a while ago. I'll need to update it. Unify it. That's what I'll be doing in the nearest couple of days.

Also, a funny thing. If you send a request to the database "give me a list of business units" in the case when you specify that you need 100 business units. Any other nonnegative number is ok, 100 is not ok. 100 is a "default value" for them, and I got a feeling they cheat somewhere. Since they don't have tests, it's almost impossible to figure out. A special kind of engineers.

May 03, 2022

It's after 10pm now, now what.

I've closed a couple of cases. Or maybe three. Just one is left, pretty fucking complicated.

It turned out that about three weeks ago, in a rush (we were fixing a production failure) had temporarily fixed a bunch of tests for this service. Now it doesn't work.

I've changed the code (switched to use Eric's shit). Uncommented the tests. Spent the whole evening trying to make these tests pass. So far only a half of them pass. But something bad is going on on the server, no changes.

(sigh) Will investigate it tomorrow. Currently end-to-end tests are throwing NPEs, because this fucking Eric, when a JSON that he receives from the server does not get parsed, he returns fucking Null. That's what he learned back home in Beijing, when he was learning C.

Oh shit. Now try to tell them about monads.

Let me start the build and the deployment now, and I'll go to sleep.

May 04, 2022

Not much to say today.

1. Spent half a day investigating, WTF is going on here. Meaning, how come the schema is not getting updated on server restart. Well, added logging, and found a couple of problems. a) our fucking Eric, instead of updating the schema for each project (meaning for a bunch of tenants), for which a request arrives, updates it only for those that are present in some configuration of his; b) Eric's code that is supposed to do this update (I have added a call for it, in just one service on one server) does not do any updates, because it just finds that this data type is already present in the database.

Nice. I created for him a case in Jira, writing that it's blocking for my functionality, which is blocking for Jordan, whose feature had to be released on Monday, and they were already asking me if I can cherry-pick it. I can, but first let Eric fix his shit. But he's pretending to be busy. And I'm not busy, having spent two weeks investigating this crazy behavior.

The other case that I have on Jira (in the style "make us a new entity, with all the whistles, but so that it deploys on another server) I started cutting it into six pieces (I'm from St.Petersburg, what do you expect), and sent the first part to Bill, who approved it right away, after which I wrote another piece, adding what was missing, and simplifying the structure... in short, it's 24 files. But so far everybody's asleep.

Not it's half past nine again; but imagine what a pleasure it is, working when it's dark outside. I'm sitting outside. A couple of ours ago it was hot and humid like in a bath. It's okay now. Frogs are screaming, one to the left of the pool, another to the right. And some kind of bird sings along.



Here it is, this frog.

May 05, 2022

Yesterday filed a bug for Eric, but he's a fucking idiot, isn't he? He responds: "you have to learn how to debug". Wow. Meaning "on my side, everything's correct", etc. I notified the management about his response. Because, really, wtf. Just another such case, and I'll go complain to HR. Well.. can I trust the HR? Funny, right. The HR is not (probably) Eric's relative, but still. We will see. Last time I encountered this kind of rudeness (from the same kind of idiots) at Google, and that's it.

Whatever. I understood that he's not going to fix anything, and I have to a) figure out his garbage of a code, b) find a bypass. I'm doing both of these.

Had a 1-1 with Vignesh. We agreed that I'll write a coding style guide. Because our veterans, 7 years at the company, don't seem to have learned anything, unlike our CEO, the Wise Leader. He's relatively

not bad; if he has mental problems, that's not much, the only culture, "you have to listen what important persons say" is coming from him. Eric has problems with that. He could do more bullying, but people don't take him seriously. People ask him: "Eric, do you know how to write tests?" (He doesn't.)

People are in panic: the functionality has been broken for the last two or three weeks (they were not aware, just Jordan), and now I started fixing it, and now it's "very urgent". Hmm.

Since I don't have much else to do, I just work. When my wife arrives, we'll be able at least to go visit places around, and, in general, I'll be having fun. Because right now I'm responsible for the whole module, Payment Integrity. That is, for its backend code. We do have serious professionals, and Jordan (scripts) and Bill (nice talks), and Bo (sullen Swedish frontend guy), and Santosh, a brilliant QA.

May 06, 2022

Spend the whole day on some kind of nonsense. I was looking for a way to synchronize the database schema. If they were not giving me advices, I'd have done it, but now, too much noise around.

I ask on Slack, how can I make it so that this config file in /opt/share/conf is not overwritten at each deployment? Actually, I just reassigned this bug to Eric. And he writes there "you need to change one yaml file", and reassigns me the case. We all know that reassigning jira cases is considered very rude. I'll be complaining to Vignesh, probably. I found that file. I changed that file. Then spent half a day beggint our devops, or, rather, "core engineers" to approve this change. Got it approved. Merged it. Now what? What do we build? What do we deploy? Silence.

Meanwhile, Jordan is in panic, Santosh is in panic, Vignesh is in panic. Meaning, Monday they are going to deploy to production, but tests are failing.

There were no changes in the code, actually. The problems are in the config files (and in the logic of updates, it's Eric's code).

Okay, will check it tomorrow, what can be done. I got an idea, store everything in the code, and keep the schema there. Because, with this people, it's impossible to get the schema, it's all boarded up, Chinese style.

And David asks Eric, in a public chat: "Eric, what did you change, I have an NPE". Nice, fucking nice, it's just a kindergarten, and everyone is such an alpha male, at the limit.

Well, whatever. When I was having dinner in Kounter, got a message from Vignesh, on Slac, where he tells me that since this "bug" is a P0, I have to write a status report every two hours. Wow, can I get some sleep at night? Or are we now in the trenches, and there's an enemy attacking?

Actually, it's all nothing. Don't consider it as complaints. I don't care, I'm in the theater. Vignesh too, by the way. We are dancing Indian dances, we smile and follow the rules of the dance. That's the right approach.

May 10, 2022

In the morning I was under weather; wrote to them, and that's it. The bug is fixed anyway, the fix is on dev. Staging would be the next phase.

Had a meeting at 1pm; I reported, saying, this and this, now I'll work on staging, but first let's figure out how to test anything on staging. Bill told me that he knows the trick, so we agreed, when he has time, to get together.

Then, at 2:15pm, we had a "big meeting", with all kinds of buzz, and eventually got to my P0. The people were buzzing a lot, but in the end Prasanna asked, how come this bug is still on staging, for the whole week now. Well, said I, it wasn't working on dev either, what's the point fixing staging if it doesn't work on dev. Told him this story, that if a customer adds a project, *somebody* has to send a pull request to devops, saying that, see, a project was added to *master*, so please; then *somebody* has to merge it manually, then *somebody* has to run a script, locally, on dev, on staging, in Production; and then I added that on Saturday I have canceled the necessity of all these undocumented actions, and the schema synchronizes when necessary, in real time; no big deal, when the code has this functionality. Eric tried to interfere, like, 100 times, to tell us what's the "right way", but how are our customers supposed to do it, is not clear at all. After which I had nothing to say, because now Prasanna started teaching Eric to develop software. Explaining to him, why there needs to be no "configuration" if we can figure it all out live, in real time. We are programmers, not accountants. Etc. Etc.

In short, Eric hates me more and more, every month. As if I care. They can kick me out, I'll find another job, big deal.

Okay, and after the meeting Bill showed me a class (which he, as he told me, had dug out a year ago), via which one can run tests on Staging. Cool, that's it, I thanked him, told him I won't bother him, and will inject it all myself.

Oh, and there was still a question: how exactly can we run tests on staging? How do I deploy to staging? Meaning, I can commit code there? Bill answered all my questions, so everything's fine, actually.

By the end of the day I fixed a bug on staging, it was a one-liner. Well, it was a historical bug. Tomorrow I'm going to slap together a couple of Jira cases, one is "fix tests and clean up the code" (and understand the essence of all things); the second is "write a coding style guide for Scala (and understand the essence of all remaining things).

May 11, 2022

In the morning (having turned on my computer about 11:20) started writing a "post-mortem" to slack, regarding this incident; but as soon as I started, Vignesh popped up and assigned me responsible for this incident. Not very late, right? He gave me a link where I was supposed to post my postmortem.

Big deal, I wrote it there. Including saying that we need up-to-date docs, and the rules about handling incidents. And an adequate set of adequate tests (we never had time for that).

After that I started writing test, according to the new scheme. Meaning, refactoring. During this process I found tons of amazing things, like this idiot TestNG wants me that all `vals` have an annotation `@Test`.

Eventually I got too tired and went on my private businesses. Tomorrow I'll run a debugger in TestNG. They have total nonsense in their docs. Anyway. We'll figure it out.

That's it for today.

May 12, 2022

Early in the morning I started working on the tests structure, to make it so that the same code could run on dev or on staging, depending on a config.

The main issue was to understand, how come, if a class has `val x: String = "this and that"`, then the testing reflection suddenly discovers a setter for that `x`. And then it complains that since this “test method” needs a parameter, it needs an annotation `@Test`. In short, tons of weird nonsense.

As to generating a setter, I'll figure it out separately; to prevent “testing reflection” touching this “method”, it seems to be enough to declare the method `protected` and change `val` to `def`.

But here's what I figured out. Our “core team” is totally incompetent or crazy or both. They declared all end-to-end tests as traits, and slapped together a class `DevEnd2EndTest`, which inherits all these traits. I remember, in around 2011 a similar trick, pancake or something, was popular in Scala, but it was not about slapping the whole application together out of tons of traits. Cargo cult, I guess.

In short, if your `TestA` has a method “`testDelete`”, and your `TestB` has a method “`testDelete`” (for very distinct types of data), one overrides another, and you'll never actually run the test cases you would need to run.

So, the trick does not work. How can we do it, after all? Call “all test methods of this class” from another class? TestNG does not give you such an opportunity.

So I started learning TestNG. Watching a pretty hard sequence of videos on youtube. Why are they hard? Because the English of that Indian guy is horrible. “Soot”, ok, it's “suite”. But other words...

In short, it's pretty simple. You create a “`testng.xml`” (any other name would do), and, in IntelliJ, just “run it”. Tomorrow I'll figure out how to do it from mvn.

The other question is how to pass the configuration name - easy! You can specify parameters in your `xml` file, that's it.

Ok, so there. Also, made tests parallel, so instead of three minutes they run in about 7 seconds. Profit!

Oh, and other funny things. Yesterday somebody (in their “end-to-end test”) deleted the account, on dev, that all end-to-end tests were using. Is not it nice.

But that does not matter; I’ll create the infrastructure, and then make sure things work.

Oh, and I had a 1-1 with Vignesh. First, we talked about work time. He was patiently explaining to me that if there’s a P0 failure, we all contribute very long hours (I won’t ever). For example, in 2019, Salesforce was down for 5 days. Because of an error made by an Engineer. He was that Engineer’s manager. He was not sleeping 6 nights (that’s obviously a lie).

But bingo! Now I know why he’s not at Salesforce anymore, and what he learned from that incident (nothing).

Next he, strangely, started to convince me that working on End2End tests infrastructure is less important than new features, because we are a startup.

Do you think I argued with him? I mentioned that we’ll be having more P0, but it’s okay, obviously.

Maybe it is. Maybe that’s how he’s trying to demonstrate his importance and his ability to work on making the team work hard.

I actually don’t want to work hard. I like to work, but I prefer that computers work, and I just tell them what to do. He too, obviously, but he wants the team to work...

Anyway, it was funny. Will our company go public? Not this year, I guess. Nobody will go public this year.

May 13, 2022

Trying to be short.

Since I’ve learned now to configure TestNG, I’m pushing almost to production. All this is not without hallucinations, but it’s manageable.

A good news came in the morning: Vignesh Has Canceled Scrum. Wow, so, there will be not anymore this senseless waste of time in listening the Frontend People speeches about which button causes which reaction, and which reaction there must be, and where to insert a form. I remember trying to talk about this to Arie, but Arie already didn’t care, so it was rolling the way it was rolling. People are happy. To be more precise, me and Bill are.

Next we had a meeting on the topic of “Sprint Planning” - it’s like scrum, but lasts for an hour. Actually it lasted for an hour and a half, and now Vignesh is writing to me asking why I’m not at a meeting. I’m responding him that I am at a meeting. And he writes: another meeting. I was not invited, I answer. So he invites me to another meeting.

BTW, at that “Sprint Planning” meeting, my name was mentioned more than anybody else’s. Not in the sense that Juan broke everything, but in the sense that “we have this feature, and we have this API, and we need to change this entity, and also there’s more” - and every time it’s me who’s going to do that. Seems like I’m the only programmer in this project. Which is convenient. I just send a pull request to Bill or Stephen, they don’t care, they approve it within three seconds. No impediments, I can just work. Okay, cool, I’ll organise the code. Starting with the tests, because how can one refactor anything without tests?

Meanwhile, at the other meeting there’s some kind of review. They were planning to listen to my talk, first of all, because, no kidding, there were two P0 incidents, although pretty insignificant ones. But since I was not invited at the beginning, the plans have changed, and they listened to other gentlemen.

Right after that we had an all-hands meeting. But I couldn’t: it was 3pm, and I’m hungry, and I have no salad. So, I quit, and drove to the nearest Aldi, bought some salad, and some fresh salmon, and some tomatoes.

I returned by the middle of the meeting. The slide on the screen was like from “Pravda” (“The Truth”) newspaper: “cultivate a sense of ownership in the employee”. In my view, it’s easy: increase the salaries and grant options. But that’s not what Prasanna meant.

Comrade Prasanna was giving a beautiful speech about initiative, and the like. Then comrade Eric gave his speech; he covered the issues and demanded more active position from the working class. Strangely, nobody applauds at Zoom meetings, everybody has their microphones off. Of course, if you turn it on, everybody would hear kids screaming, dogs barking, wife grumbling, contractors knocking, drilling and sawing (that’s what Bill has in his new house). So, no applause, we just silently smiled and nodded, demonstrating our happiness about our work progresses.

Then, having taken a short break from this nonsense, I sat on my porch with my work computer; I was sitting and watching the rain drumming on the pool, the lightning sparkling behind the woods (I was planning to go take a walk, but no), and, quietly, I was civilizing these end-to-end tests.

Now I’m pondering, should I stuff the configuration for different platforms in Yaml, in JSON, or just into the Scala code. I bet by now you know the answer. I decided to put it in the code. This way one can introduce a common trait, and there will be no dependency on unreadable data. So there. Will go work more. Ok, it’s 8pm now... well, ok, just half an hour more...

May 16, 2022

It was a good day to work.

On the scrum meeting (yes, we have scrum again, an hour-long, and, since, orthogonally, another team was formed, that other team as scrums consisting of just daily reports. It’s more convenient, of course, you just send three lines, and you are done.

At this scrum our pm made me happy, informing me that there's a Slack channel where they assigned to me yet another task; I told him I'll do it with pleasure, but last week we had two P0 failures, so I decided that it's most important to create an infrastructure where our tests would run on dev and on staging machines; but if he thinks this is not as important as a new feature, I'll gladly switch. Now he got in denial, saying that, oh, no, reliability is also very important, but could you... etc, etc. Of course, - I reassured him, - I'll get to it, as soon as possible.

The next one, the “big scrum”, the usual on Mondays, lasted for two hours, and everybody was reporting to our dear Führer, or, rather, reported and then respectfully listened to his opinions regarding these achievements. I was too fast saying that well, I worked on two P0 errors, and, to prevent their repetitions, I'm doing all I can.

By the way, our test account was broken by some of our heroes, so... so, no problem, Jordan told me it's not a problem at all, and sent me a link (I could have guessed myself, but I didn't); in short, not a problem.

But now it turned out that half a dozen of my end-to-end tests now fail, and why? Because Eric's and David's code now does not convert the old test data in the database into instances of a required class, and why is it? I did not touch that class in the last month. But since our logging is totally shitty, I'll have to first fix the logging. Since even a 7-years-old class does not have the `toString()` method, and all we can see in the log is the address of the instance. Of course, HIPAA is involved, we can't show everything in the logs (can we keep it in the database?), but I have no plans to walk around convincing them; instead I just slapped together an implicit class, in the style of pimp-my-library. So there.

Since Stephen is on vacation, the whole week, I merge it all myself (I show it to Stephen, but he's on vacation), at the same time injecting a little bit of my improvements (improvisations?) from two other branches, it's just tests and logging. It's not a good time to change algorithms, first we have to make sure all tests pass.

That's it; it's half past nine now, and I'm closing my computer. Ok, I'm taking breaks during the day.

May 17, 2022

Having fixed logging in BU service, I have merged it and deployed it (these are just tests and logging, and we agreed with Bill that there's no need to review tests), and then I wrote to Santosh, saying, hey, Santosh, take a look, please, will your test pass now? (My tests have been passing for a couple of days now). Suddenly Santosh's tests are passing, so I closed the case.

Generally speaking, a bunch of end-to-end tests are failing for me just because our fucnkign idiots represent structures in the database in a very strange way. E.g. JSON with no quotes. My trouble here is that I don't have yet a method to delete these BU records. Ok, I'll make it later. We do have problems. Later, later.

Now I started configuring the tests, to make them work on a bunch of platforms. Finished configuring them by the evening, wrote to the people, like, beware, I'm going to run tests on staging.

Meanwhile, they are bugging me with something like Khiron, or whatever, they say there are tasks for me. I'm asking them:: is there anything for me in Jira? No, nothing in Jira, I'm supposed to read some docs. But the docs are not written yet. "Stiffen" (as some of my colleagues call Stephen) is away for a week, and nobody knows what he wanted from me.

That's how it is. I nod and smile.

And, you know, I have other things in life, in parallel. E.g. my book...

May 18, 2022

I was busy, working all day. Eventually I made the tests run on staging, but now it turns out that we have to organize it properly.

Will document it tomorrow.

That's it for today. I'm just busy.

May 19, 2022

Nothing today. There was a feature that wasn't working; I've fixed it. I was keeping the case open until end2end tests were written; wrote them, discovered a bug, fixed it, but the tests were still showing errors, so I fixed the tests.

Now I could close the case.

Then I wrote a doc regarding how we can better organize tests on both platforms; send it to the people for review; Bill added a good comment, but I also wanted Jordan to tell me which project is better to start with. But Jordan was busy, he filed me a bug. It's not my bug, it's a known bug by Eric; but Eric will be inventing all kinds of excuses; and we want the stuff to work.

I'll look again tomorrow, how can we fix it. Tomorrow.

Also, I had streamlined some code. Gradually getting rid of all that crazy shit; the code is becoming simpler and shorter.

Then we had a meeting. Vignesh had canceled the scrums, and Anil had reintroduced them. Bo manages to participate in the meeting and work simultaneously. I can't do it (why?), so I read blogs while they are buzzing, for an hour or more.

May 20, 2022

Nothing interesting today. Yes, I'm still trying to slap together a pull request - but something's not building, sorry.

May 23, 2022

Three hours of meetings, the rest dedicated to work. In the morning Stephen approved my Friday's pr, so I merged it in, deployed, ran the end-to-end tests. Ok, the new functionality works ok, but there's a bunch of failing tests.

The second meeting was when our Führer was asking everyone, how are things. He saw that I have "write a coding style guide for Scala", and said that it's a waste of time. Wow, I got it. We have a hacker for a CEO. Ok, the company is not going public this year, and hardly next year; the company is not growing, so, who cares. I care about my own style, but I know nobody cares wtf I'm writing anyway.

Another issue mentioned what was the failure that end-to-end tests are reporting. Well, things have changed. Eric asked me when and how I'm going to handle it all. I told him, I hope tomorrow I'll file a ticket for him, since it may be in his code. Silence. Wow, he did not start screaming. Hmm.

When I had time, worked on the failing tests. By the end of the day, after totally distilling the failures, came to a conclusion that Eric's backend; so I'll reproduce it and file. Mañana.

The weather on my porch is beautiful, that's where I spent my workday; it rains cats and dogs, my pool almost overflowing, but not yet.

May 24, 2022

In short, the bug turned out to be mine, not Eric's. I traced it, wrote a unit test, fixed it. How come the unit test was not catching it? Well, my mock service was too kind. I had refactored the service, but the mock remained the same. Ouch.

That's it. Then I was busy with all unrelated stuff, I was driving back and forth. Meanwhile, on the road, I was thinking how the new service should look. It turned out simpler than I thought. Okay.

By the way. At today's meeting our PM and our manager were energetically discussing race conditions; our ui guys joined the show. I was silent, I know where all these topics come from: from our Great Führer... in short. let them talk. What exactly race condition was it, they did not specify... actually, they were clueless. It's just a great idea, blaming race conditions for any bug in the code (there are no tests); it's just God Almighty deigned to mix up our cards. We spent about half a year tweaking our "tables", to build a "strongly-typed API" between ui and the backend. Some kind of "Entities" were introduced. And now our Great Führer says that we don't need "Entities", we need UDT. Wtf is UDT? These are User-Defined Types. That is, JavaScript dictionaries, I guess.

P.S. Maybe Prasanna was right.

May 25, 2022

Something is not right. I wrote a new server, as a prototype, but I can't send it for review: `git push` just hangs. WTF.

In another thread I have another trouble. All the tests pass, ok. I run the tests again, and one test fails. I run a pretty simple test case: "create a record, read it, compare it, read a list, find it in the list". At times it is in the list, at times it isn't.

Now I'll have to work on Eric's code.

And yes, `git pull --rebase` hangs at every other attempt.

Meanwhile, our managers decided to solve the race condition problem. Sure, why not? I should also familiarize them with the halting problem; that would be revolutionary.

May 26, 2022

I was messing around with two jira cases, all day.

One case, where I introduced a new service, it was kicked out by Eric, because some beans were not being instantiated during the deployment; well, whatever, I'll figure it out.

Another is a mystery. You store data into the database, the database takes the data, but it does not read it. I had reduced it to a unittest, and now I'm messing with Jackson, since if you rename a field (in a certain way), it works, and if you don't, it doesn't.

WTF, right? It's already 22:22, and I'm still having sex with all this.

Had a 1-1 with Vignesh; we exchanged a couple of nice words and closed the meeting.

P.S. Well... now I have a suspicion that, with our backend, Little Bobby Tables can destroy all the data in one fell swoop. From the client. P0, I guess. Ok, I'll figure it out tomorrow. Nobody knows yet.

May 27, 2022

So far the problem (problems) is not (are not) solved, so there's not much to write about.

In short, we have incompatible serialization and deserialization, and a crash due to their incompatibility. Working on it. And of course the effect of Little Bobby Tables.

At the first meeting we had I had explained the problem, in short terms; at the second meeting they had no questions to me, and at the third meeting our management told us that an economic crisis is

coming soon, but we'll get through it with flying colors. Meaning, yes, we have customers, and the customers are getting richer and richer, and we contribute to this.

I'm pretty happy that nobody interferes with my work. At times I here something like "that shit that Juan made, it allows us now..." - etc. Thank you, I like to hear this.

May 31, 2022

Spent the whole day having sex with the whole Zoo (not the people, the code).

By the end of the day all unit tests started behaving; I have kicked out a bunch of bad methods (but not all), leaving only good transformers.

Have to ping Stephen again, to ask him to kick out his use cases. My "implicit transformers" do unify the transformers. Eric has a bunch of unnecessary methods, but we don't have to know them all.

Sent it for a review. Yaron, our new QA, asked to send this pull request to him, so he can get familiar - and I sent it to him too.

Anil created for me a new Jira case, like "write a document"what to do if during a server deployment the database doesn't get upgraded""". I know what kind of document will cause a positive reaction. Write three variants of actions (or even four), and suggest to have the selection specified in a config file. And that would be it. Then let the devops make their choice, the way God and the Management tell them. And we, programmers, in the case things don't work, just report to the logs (and to the client):"oops, something's not working).

Oops, a grasshopper just jumped on my screen - and flew off right away.

At least I'll have a break. I spent four days cleaning this module, for which ass (backend) I, in my view, am personally responsible.

Last night, about half past eleven, I asked, on Slack, what are our policies for naming fields and columns. Only Prasanna answered. His answer was: "YES". See, I meant to make sure that nobody would further complain that everything's wrong. Now they won't (I hope).

June 2022

June 01, 2022

It's 9pm; I'm finishing my toils (just one e2e test is left), and I'll send it for a review.

Checked out in the morning what Bill and Stephen are writing to me; they made half a dozen of good comments; I've fixed it all, merged it, and now almost all e2e tests pass, except just four. The problems were in the tests, not in the code.

At this moment I started a script that will create a branch, run all the tests, commit, push, open a page to create a pull request (I still have to manually choose the reviewers).

Nothing else today. We worked, and we are done now. And the code is pretty neat and clean. I like it.

There was just one meeting. Anil wasn't there, Vignesh was replacing him. Everything went fine. Not an hour, like it was with Anil; just half an hour. It's a progress. Ok, will work on this new service tomorrow. It's probably not a good idea to look at other people's code. They write too much trash. Things are much much simpler IRL.

June 06, 2022

So, briefly.

Got my new service (but not attaching it to the system), sent to Bill for review; he made a couple of smart remarks, which causes a couple of fixes (like, removing the method he suggested to rename).

Merged, but did not build, because another build was running, so I just queued it, and did not queue the deployment.

Switched to experimenting with end-to-end tests on staging. Config issues, no big deal.

Then got back to Jenkins. HTTP 503. WTF. I ask devops, and they tell me that Jenkins will be down for an indefinite time (or maybe forever?) Something like that. Guys are having serious problems, and they don't have time for Jenkins yet. Wow.

Ok, so I somehow made all end-to-end pass on staging, and happily informed the colleagues on our slack channel. Prasanna was the only one who reacted. Properly.

Ok, then we had a scrum meeting (pretty boring) and a big Monday meeting (boring too, but I at least was having my lunch meanwhile).

When my turn was for a show, I (as I started doing lately) made a show of it, how end-to-end tests detected a failure in core immediately after it happened (and Ananth interrupted immediately, saying, ha, that was my bug!), so, in short, we together had just entertained the public.

And then I was looking for solutions how to reproduce tests successes on staging, because they fail every other attempt, and succeed every, eh, another attempt. Weird. Have to figure out. Also, have to make sure that when I rerun failed tests, they rerun in the same environment. Tricky, but doable, locally.

June 07, 2022

Nothing was happening today, for a couple of reasons.

First of all, our Jenkins is still down. As it turned out, Friday 13th one of our devops had opened this service to the whole Universe. Immediately, some smart Chinese hackers came over and started mining something for themselves. In less than a month our brave devops noticed that this wall does not exist; they stopped all the Jenkineses and started investigating. It's been two days now that they are investigating.

I also must note that the whole company's specialty is internet fraud.

So, we are presently sitting quiet.

Meanwhile, Vignesh got a bright idea: let's write good docs for everything; come up, guys, with your ideas! I did, I came up with an idea, but then I fell asleep (I don't know yes, what kind of cold I just got).

When I woke up, I could not even run tests, because the dev server was also closed. So... so what? I'll get better tomorrow, and will be working on the docs. Anyway, look how businesslike is Vignesh! I like it.

And if the company crashes, I'll go to another.

June 08, 2022

It's quiet at work today.

Except that Jordan asked me to add one more value into one enum. So I did.

Otherwise, they seem to have installed a new Jenkins, and I don't have access to it. (Well, who are we, after all.)

But Sebastian was nice today, and created a task for Jenkins, to run tests there, and wrote a script, which I had fixed, and lo and behold, we can run tests. Except that I have no access, to run unit tests in Jenkins.

Oh, by the way, I don't have access to the dev machine either; I wrote, in the morning, on the channel "server team", asking, hey, what's up, I can't access dev machine, will it be fixed soon? They respond:

“what happened?” (as if I haven’t just describe what happened). I posted a command line with curl, demonstrating what exactly happened. One guy answers: “don’t write stuff like this anymore”. What? I don’t get it, what happened? Is it that I mentioned a session key on Slack? This session key is in our repo, in the code, and everybody can read it. What are they suggesting? Not to test anything at all? Just not to be afraid of risk? Well... FIIC.

That’s how I finished my work day today, with zero results. Not exactly; they are paying me my salary, whatever the outcome.

June 09, 2022

Although I did not do anything “physical” in the morning, we just went for a walk, I was still sleepy and tired. So, I’m sick, or something.

There was a 1-1 with Vignesh, but I was asleep.

But I managed to do something, like, a) fix pom.xml, so that Jenkins could see the test results; b) access the pages of the new Jenkins’ c) and now unittest do run, officially, during the builds (not to the degree to crash the build if tests fail, but still).

Also I managed to deploy this new service. But it returns a 404 for some fucking reason. Why?! I don’t know. Have to investigate. So far I can’t access our dev machine via ssh, that’s it. Pinging our devops.

Also, around half past five I had some kind of interview with a pretty interesting shop, split from Galois; they have DARP etc. I got a feeling that it didn’t click: the interviewer was almost yawning. Well, I won’t suffer if it does not wok out. In our hospice they at least don’t interrupt my sleep.

June 10, 2022

Too lazy today.

Well, see, first I had asked “the people”, how, in general, are we supposed to react to a bad request, 400 with a text or 200 with JSON? Opinions varied. So I decided not to change anything. And started tweaking this new service.

At the meeting Bo solemnly showed his covid ticket with two bars. Pregnant? Kind of. He has covid, and Stephen has covid; Steven blames his elder brother (I think he got this habit since early childhood). I have undetected covid. No, I’m fine today, actually.

What did I do all day? Hard to tell. First, learned from Alex that now the dev machine has a different name, and the other dev machine does not exist anymore. Then he confessed, and told me that there is another machine, called postgres-1. This is nice, isn’t it?

So, probably, I can go ahead and investigate all this bs. Oh, and my ssh keys have expired, and I had to generate the new ones. I need to write a script for that, instead of remembering it.

Nothing else. The work continues.

June 13, 2022

So, in the morning I prepared a pretty neat PR, about 20+ files, sent it to Stephen. I was proud that I found a generic solution for an entity factory object; will be expanding it.

Meanwhile, my service was misbehaving.

So eventually I asked the people. And David pointed out that, ha, I wrote with my own fingers that the user has read-only access for that method (and needs a `create` access). Omg. That was easy.

So, I wrote more stuff, and by the end of the day sent a PR to Stephen and Bill again - and checked slack.

OMFG, there's another P0 incident. Somehow they tried to involve me, because I was changing the accompanying object to the class that failed to sync with the db schema. The only thing is that the class was actually changed 2 weeks ago, and it failed Sunday, before I committed anything. So there. It's "Stephen's area" - but actually we *should* eventually take care of db schema synchronization done properly.

So I offered help to Stephen, and shared some philosophy. I know, nobody fucking cares about my philosophy; that's natural. The guys never stop and think, they just rush to fix something immediately.

Anyway, that was a good cause for me to close the work computer (ok, it's 21:32 now); I'll open it again before going to sleep, but they can't expect from me spending half a night working on Other People's Problems.

Oh, and, as Anil asked, I created about four Jira cases today (and also he asked me to slap together a design doc for some shit. That's not a problem for me, it's even easier than programming.

At a meeting today Eric managed to convince me that logging each http response is not necessary. I figured I agree. Clients may log it. Security people may log it, but I'm not sure, we have PII and should follow HIPAA. In any case, I figured that at this stage I don't need any response logging. Responses arrive, and the test may always figure out wtf is there, so what's the point. Funny. Now it's obvious. It's like "`println`" in tests.

Meanwhile, they found that the failure consisted of inability to convert an int to a double. I kind of know where this comes from. Not from Haskell. From Ananth, who's been playing with conversions, without bothering to add test cases. Anyway, David said, since we are having troubles with schema updates, let's stop making schema updates. (My ideas about testing the code regarding updates before making any updates were, as usual, ignored. Which makes my life easier.)

June 14, 2022

Busy like hell or something.

So, the guys were still working on p0; blamed Stephen, but the problem is, Stephen last time touched the code on May 2nd; and I touched it *after* it started failing (if I only knew).

So, a bunch of unrelated people got involved. Eventually David and Eric decided that the problem is the order of fields in a database record (surprise). Who cares about the order of the fields? We are not C programmers in 1990. Anyway. Somehow the management managed to convince Eric that field order should not matter. He was just, meanwhile, trying to convince the world that the order was wrong, that's why, etc.

So, nobody had time to review my pr, and I just merged it (ok, I had to create a new pr, because github said it can't be merged).

Anyway, my tests fail, and I figured eventually - the thing needs a "project", that is, a namespace. I do have a project in my service, but the guys use some magic to set the project "in a thread-local something"; so I asked them. David pointed me to a code sample; OK, works for me (not yet; pushing now, but well, I got the idea).

Meanwhile, slapped together a script that takes jira case number and a comment, and creates a branch, compiles it, runs tests, pushes to create a pr (creating a pr is next in my plans). Had to use tr, to replace spaces in commit comment with underscores to create a branch.

Ridiculous, but I'm trying to automate as much as possible.

Have to finish this part Thursday, so I have two days left. In any case, tests and the server part are promised for the next week, so I have a cushion (as if anybody cares). Prasanna asked me today at a meeting, who is going to use this new service? I said - nobody, but Jordan is very interested in having it.

Oh, and they asked me to write a design doc; so I started; the basis for the design doc is some babble from Prasanna that I could find in slack (Jordan provided me with the link to the head).

June 15, 2022

The best way to get rid of an "important task" is to add into the document questions for the PM, asking, like, tell me, my friend, where exactly can I get these data? Where do I send the results? What's the meaning of this line?

Then silence falls: the PM is just too busy with other things... and then... a week passes...

But the same PM usually has lots of time for multi-hour meetings, just lots of times.

June 15, 2022

Nothing special today.

Jordan has filed a bug that in one (side) project one record type is not being created, complaining that such and such field doesn't exist. Ok, I tried. In the common project the record is created ok, but the problem is, I don't have access to Jordan's project. Well, what's the point of fooling around? Yes, we are all on the dev node, we could keep it simple. But... but things seem to be more complicated than they could be.

Whatever. I'll check it out again. Maybe I could just read the logs. Grep them. I have my scripts.

BTW, talking about scripts. This morning I was merging something to my branch, small updates - and kaboom, the branch turned out to be master, and I had merged to master right from my mac machine. How t.f. is it even possible? WTF, right? Who allowed this. Well, I fixed my script, started checking the branch, but... Asked them in the server channel on Slack. Sebastian answers: "you need to change your git config". Write something like "simple" in the config. Meaning, that's how we do things, and I had to read all the comments to comments, for the last seven years, to learn the truth. Ok, big deal, added this recommendation into wiki on github. what do you expect. By the end of 2025, I'm sure, they either deleted it or forgot it.

That's it for today. I promised to finish working on a feature by tomorrow. It does not work yet, and it's not in the plans to make it work; I will be doing the server part next week. So, I implement it formally, and then I'll work on access problems, I have time for that.

June 16, 2022

Basically, was busy working. As promised to them, implemented all methods. No, they don't exactly work (permissions, database schema etc), but they are formally there.

While experimenting and fixing, did a bunch of refactorings, pretty neat ones. Simplifies entity management a lot.

Also there's a bug filed by Jordan, about db schema not being properly updated in a new project. Strictly speaking, it must be Eric's bug, but well. So, when time will allow, I'll try to reproduce the experiment, and see what goes wrong. Logs from Monday are already invisible to me (if they still exist).

Had a 1-1 with Vignesh; nice as always.

At 4 there was a meeting regarding on-alert duty for our team. Bill was trying to convince the management that we won't be able to produce software while on alert; Jordan said that this will help us find bugs and report them and improve the product; I said that it's our product and it's our responsibility as of now; we will see later.

Worked until about 8pm our time, improving, refactoring, gradually converting instance-based code to type-based. When yesterday I found a way how to declare/define factory objects (accompanying objects that can produce instances etc), it opens the gates for civilizing the code base.

June 21, 2022

A month ago I created a Jira case, assigned it to Eric, asking to write unit tests for this code.

Today, at the large meeting, Eric was wriggling like a snake, and eventually Vignesh got an idea: assign it to me, write tests for Eric. I said something like “fuck no, I’ll never write unit tests for other people’s code. Unit tests should be written by the code authors exclusively.”

In short, the topic was closed. They’ll be looking for volunteers to write tests for Eric.

What kind of bitches, really.

Well, whatever. Cool down, ignore them all. They are not cooperating.

June 22, 2022

I got a feeling that I didn’t do anything today. But actually my yesterday’s pull request was approved, by Stephen and David. And I had merged it. And the build crashed, because, fuck, Eric has some tool that “verifies” Scala code, and we here can’t run this tool. WTF, really. Asked David, how can possibly run it? Meanwhile, Prasanna, on Slack, asks whether I don’t run this tool locally. Well, this is the first time I see it; they don’t show us these scripts.

Fuck. In short, I’ve fixed the bug (it’s not a bug, it’s a warning in a test, just nothing, and there will be just an exception, exactly what I need). But I started asking around, how can I run it locally. David gave me an idea; and I configured the build... but it does not crash for me, locally. Because it analyzes build logs, and how exactly do they build it in Jenkins, not the same command that I run? Then which one is it? I have no access to see, how do the scripts look like. Not bad, eh?

Then suddenly David left his ebony tower and wrote to me that I have many errors and warnings in my code. What?! Where?! How did he reproduce it? Wrote to him. Silence. Warnings... these are not warnings, these are errors. It wouldn’t have compiled. But it was compiled and deployed. What now? Show me the errors. Silence. Seems like... he might have a very different environment. Or what should I run? mvn clean deploy ? But let him show me the errors he’s observing.

Otherwise, everything’s fine. Everything’s just fine.

By the end of the day I managed to convince David to do ‘rebuild’ in IntelliJ, and all these “errors” have magically disappeared. More, his idea about Integer instead of Int, because Int is not nullable, seem to be as meaningless.

(How tired am I of all this nonsense, omg)

June 23, 2022

So, this morning I got on Sebastian's neck asking him to show how exactly should I run the build to reproduce warnings in the logs. And another question, how to set up (programmatically) proper access to a new entity.

Since the guys were busy with their P0 failures (job security!), he did not answer until after lunch; and the tone was... well, you know how the Chinese talk these days with those who are not their managers. Same as Russians.

Anyway, he pointed, at least, to a document somewhere in the pile. I'll read it.

Had to have a 1-1 with Vignesh, but he's busy with the P0 (the best occupation, watching other people working, and driving them nuts by giving them a wise advice). Anyway, just agreed with him that we need a culture, not fire-fighting.

I slapped together a big pull request, whereby I removed all opportunities to subclass a case class (it's legal, but results may be surprising), sent it to Stephen, he approved it. And so on.

Spent the evening slapping together a test/build conf that would react to that warning that we saw yesterday. It's not 100% trivial, but getting there.

That's it.

Boring? Not for me, I'm having fun, intermittently diving into my pool.

June 24, 2022

Okay, I managed to make this stupid Eric's test run on my machine; I have a script for that, so everything's fine so far. Later I'll have to publish it in github, for the people to use. But somehow Chrome browser was misbehaving so badly that I decided to reboot.

Ok, and I rebooted. Our devops made us install a new version of Chrome, with some new "security features", green-blue color; my whole history was successfully deleted, together with all my bookmarks. Good job! Now, how can I remember all this?

I can't even log in to the calendar. So I want to the meeting via my phone (one needs a microscope to read letters in Slack, and I have no microscope).

Ha, now I remembered, I have a reference wiki page in github; these idiots could not erase that page. Otherwise, I don't remember the names of the scripts via which I was connecting, by ssh, to two dev machines. I looked and I looked. Found, but they were not letting me in to dev-0. Now they told me to try another name. And so it goes. Oh, whatever.

Spent about half a day restoring the environment. Well, almost half a day; my reference page helped. In the other developments, we had four hours of meetings.

Also, Jordan and one other guy filed a bug: if you are creating a new project, the database schema is not getting updated. And he, I already have a test for that that fails.

At one of the meetings people discussed an interesting problem: how should we stop deploying bugs? Really, how? Meanwhile one smart devops guy wrote a pull request, to prevent merging before all the unit tests pass. As we agreed already, with him and with David. Some progress, eh. But the people continue discussing the problem of tests. See, tests take two hours. I don't know why it is two hours for them; my unit tests, for our module, take a minute to finish.

I know, I know, they, for every test case, log in to the services, waiting for the services to respond, etc. So there.

Whatever. I don't want to join stupid discussions.

It's almost 10pm now, and I'm still deploying. There seems to be one nasty bug, in Eric's code... that's what I think. His code does not accept structures containing parameterized classes, except for `List` and `Map`. This is a mid-level stupidity, but the problem is that from these structures he tries to get into unrelated classes, where I keep regular stuff. So, why so? And he does not disclose, how does he get to these classes that are not related. Eric.

Okay, I'm investigating. I gave P1 priority to this case.

June 28, 2022

So, it's Monday. Wrote to Jordan, asking to verify that the bug that I was working on is fixed now, and if so, to close the Jira case. Jordan didn't check, but agreed that it's fixed.

Now I started reading all those docs regarding ACL ("access control lists"). The text is rather murky, written in the Chinese flavor of English.

Then meetings started. During all this time, I was working on the closet, cutting drywalls and attaching them.

Then another meeting, when Prasanna was pestering everybody; eventually he got to me, asking, what am I doing. Here, I said, Sebastian sent me a link to ACL docs, and I'm reading it. Prasanna says: but we talked about it last week. Sure, it was Friday evening, now is Monday morning, and? Who, by the way, is going to use this feature? I guess, it's Jordan. No, - he says, - Frontend will be using it. You must, he says, be more interested in how the system works as a whole, you must ask questions. Do you have questions? I sure do, I say, I ask Sebastian, how these ACL work, and how come they don't work for me, and how about our code reliability. These are wrong questions, he says, you must be interested

in the system as a whole. Aha, I say, you know which questions I must be interested in; that's good: so, I should not be interested in ACL? Now he started telling me that the class `Sentinel` is not currently needed; what we need is `SentinelVersion`. Eventually it turned out that he should address this to our PM: I was told "give us a `Sentinel` entity", and so I do.

In the end we chatted with him via Slack, discussing issues. Of course he can't, for some reason, explain anything meaningful to our PMs, so he shares his vague consideration right on Slack. Well, I did ask some questions, so, ok, now it may make sense. Maybe he means that I should address all my questions to him? I don't know, maybe; I'll do it with pleasure (practically he's pretty literate in software development), but I don't find it a decent idea to distract a CEO with all this small shit.

Well, whatever. Everything's fine, so far.

Meanwhile, Jordan writes (and he creates a Jira case that such and such test (of his) is crashing, with such and such error. Hmm, I have just deployed it, and all my e2e tests were passing. Did it again, ran all end-to-end tests. Oops, a bunch of tests crashes now. How come? Went to check the logs (btw, why not send logs to the tests? I'll think about it.) The logs are saying that Eric's server reports not getting `entity id`; how come? Here are the logs, here are requests in the logs, and the requests do contain `entity id`.

Added more logs and checks, went ahead, committed it, merged it, deployed it. Everything's fine. What kind of assholes is our "core team"! BTW, their end2end tests have been in red for the last couple of months.

But they are very very nice with Prasanna. And hardly talk to me. Fucking politicians.

I actually don't care at all. If not one company, there are others. There's a lot of jobs on the market. If you are not "building a career", it's just a perfect time for software people.

Ok, whatever. Closed my work computer for today. Every meeting just kills my ability to work, for about the length of the meeting. I need a one-hour break after a one-hour meeting.

June 29, 2022

In the morning I spent some time investigating that ACL issue; stopped by 1pm, with the logs open (stupid idea, the ssh connection closes, and the picture is lost - I had to scp the log to my mac).

Then there was a meeting; and there was another bug reported by Santhosh, and they said it's P1, so I promised to do it by the end of the day today.

Also, David was writing some shit, like "you guys need to write better code, you return lists in responses, and it's wrong" - I did not believe it; checked all response classes, no lists there, so I wrote that I could not find any. And David had gone postal (or ballistic), like I should not come up with this

kind of conjectures about lists not being there... on the other hand, yes, there are no lists, there are arrays.

Now what?

Then there was a meeting, me, Vignesh, Anil, to discuss plans on the feature I've been working on - my plan is growing into "milestones" (5 milestones now), and we agreed on everything. Cool guys, pretty smart and reasonable. Anil also a cyclist, rode up mt. Hamilton recently, broke a finger after a fall, etc.

They also said that David should be the point of contact for all those ACLs. While David is a little bit crazy, it's much easier for me to talk to him. He admits his mistakes, I admit my mistakes, and he does not behave, in general, like those two Chairmans Mao.

Also, these "five milestones" give me a lot of space for properly developing this functionality. Seems like they came to a better conclusion while discussing this shit with Prasanna.

Oh, funny. Later in the evening I started deploying my update to dev, and suddenly Sebastian decided to give me and advice. If, he says, you only are changing config files, use `deploy fast`. Sure, said I. As if I did not know. But Sebastian probably got spanked earlier today, and decided to be nice while talking to a dangerous insect like me. He will revenge later. Will wait until my dead body floats past him, while he's waiting.

Fuck if I care.

Anyway, I was busy with my small shit that I did for Santhosh (it turned out that the response type should be subtyping something that contains data. Well, I created that `ResponseWithData` class a couple of weeks ago. How come these guys did not bother, for the last 7 years. That includes David. Who knows. At least nobody's stopping me in this kind of development.

We will see. So far so good, actually.

June 30, 2022

I was busy all day, investigating this inadequate behavior. Wrote to David, with all the details, but he doesn't answer. I think they have a bug in the code. There are zero tests for that "core functionality".

Also, had a 1-1 with Vignesh. Since Monday I dared to dispute with Prasanna, instead of nodding, Chinese style, then Vignesh decided to teach me how to properly behave. What I should be interested in, at work, like that.

We agreed that, since "the team does not know" what I'm working on, I'll write a big design doc. Frankly, nobody, except me, writes no design docs here. They probably don't understand what exactly am I working on. I'm investigating a bug. Eric's bug. But there's no use explaining, I guess. Meaning, Jordan and Stephen, and, at times, David, they are okay, we can discuss things, but otherwise... otherwise it's just "go ahead document it". No problem, I'll document it tomorrow.

In principle, to trace the bug, we need this: first, extract from their “service” a piece of functionality, inject it into my code, attach a bunch of logging (I better not touch their shit, they’ll start screaming that I’m filling the logs with my trash), and then, then commit-push-merge, then deployment, and only then, about 40 minutes later, I can run my tests; then copy the logs from the server, and start investigating. So far I don’t believe my eyes, but at least I know where to look, thank you David, you showed me where to look. But it doesn’t work. I think, nobody knows this code, or nobody remembers. Sebastian does not understand anything for sure in this code, although his name is everywhere. I think, he was copy-pasting it.

Ok, that’s it. I suppose, in the afternoon I’ll take off, telling them that I have AFK (“away from keyboard”) and PTO (“personal time off”). Otherwise Prasanna will be asking dumb questions again, “why is it not ready”, and I’m not ready yet to blame Eric, I’m working on it.

July 2022

July 01, 2022

Well, the successes were imaginary. So I did not even want to join the meeting(s). Pinged David, Seb, Eric. They either are silent or write shit in response. Stuff does not work. The complicated part is the management of the db records; this is a crazy kludge. I took Eric's end2end test, copypasted it, adapted to my entity. As a result, the test pass, the entity is not visible. In the test, the entity has the necessary permissions, but it's not visible in the application code. So, what should I do? Write my own data wrapper? Maybe. Can do.

That's it for today. Will have to avoid temptations to work on the weekend.

July 06, 2022

Today I've been enjoying my days of delegation. So that eventually, at a meeting, Vignesh asked me if I have anything to code; I sure do; he suggested, while the Core Guys are working on the great generic feature (that would not be working for a while anyway) to implement something that works. Sure, with pleasure. Started with thinking about it.

Because, you know, the most efficient way to work is to think first.

And the best way to think is to float in the pool, like a frog, totally relaxed, looking at the wild clouds in the sky, and slapping together the documentation, the solution, etc. Then you get out and write, fast.

So I did, spent about half an hour writing everything that I know and we need, putting it in a good readable shape. Taking a break now, and will fix the solution regarding "how do I discover shared project service". NP.

Also, Stephen (our Indian colleagues call him "Stiffen") introduced "request ID" in the responses, inventing his own representation; I suggested to use the id that is already stored in MDC. Seems like not everybody's aware of MDC at all. Well, anyway.

I'll have to go to Lowe's, and then do some implementation experiments. Is not my life hilarious?

July 07, 2022

First I decided to rename my service, to make it very generic - like "SharedDataService"; but after everything, it turned out that the standard term here is "Reference", so I called it "ReferenceService" - let it keep as much data as will be needed later on.

And so on. Added creation of the project, deletion, listing, etc.

But then it turned out that those reference projects already exist everywhere, so there's no need to create it. Cool. Makes my life easier.

Then we spent an hour discussion how do we discover this shared project. Eventually they convinced me to store the ids in a config file (cheaper), and Prasanna, when I posted this discussion on slack, also agreed that this is the right solution. At least no hacks, as were suggested, and more, if you do not discover something, but have a pointer, you won't have to deal with multiple or zero answers. Cool.

So by the end of the day I posted two prs for the people to approve.

Also had a 1-1 with Vignesh, but there was not much to talk about, he enjoyed our discussion at the meeting.

He also insisted on doing the first milestone, not waiting for David to fix their bugs. Ok, I promised to continue the investigation (but in the new piece of code, I'll feel better there).

That's it, except that I got an email from Sean, a great recruiter (if you need a recruiter, I'll tell Sean); he asked if I'm interested in a company... which company? this one; and he also wrote about the salaries at Senior and Lead levels. WTF, mine is much lower. WTF. I wrote to him, thanking for the information, and assured him that I won't disclose the source. But I'll talk to the management rather soon. Next Thursday, for instance. Let me sleep on it. Spend some time.

Basically, currently I ponder on issues when I just float in the pool. Helps a lot.

July 08, 2022

It was quiet today.

I ran tests for the new service that I made, with an assumingly new project. Discovered that I have some nonsense in my test regarding resource discovery, so I fixed it. It doesn't work yet, but it's okay, I'll investigate.

Sent the configs for review to Eric, David, Seb; Erick wrote that this stuff should be a part of some script, in a yaml file, not in a config file. Well... all this is not without mental problems, but okay, I did it. Sent it to them. Eric responds: send it to the devops for review. Ok, sent.

Spent another half of the day getting through a "security training". This one is not that bad. Makes a little bit more sense. But still. But who cares.

We had just one meeting today. All the management had fucked off somewhere, either they are taking a long break, or something. So we talked business, and it was fast.

That's it for today.

July 11, 2022

So, this morning I was kind of preparing my experiments with permissions, and then we had a scrum “standup”. My first name is last in the list, so it’s convenient, usually not much time is left for me to brag.

Vignesh assured me that Eric and Sebastian left their notes in the bug I assigned to them; I did not see the notes. It turned out that Jira is an ancient beast that needs to reload to see updates. Funny.

Ok, Seb seems to be not very good with English, he was asking how come I mentioned one test case in one place and another in another place. I pointed to the word “like”: one is like another. Eric wrote that the entity I mention does not exist. I pointed at the line in the code that contains the name of the table. Why t.f. they did not bother to run the test I kind of dedicated to them?

Eric insisted on me writing just one or two lines with curl, so he could run them. Sure. If he ran the test, he could see the curl line. But it’s not the first time he refuses to look into the code. Ok. This is the last time I provide him with anything like this. Oh, yes, I added information for them. But they are in negation.

Also, another question during the meeting was, what else did I do. Updated the design doc. What design doc, asked Jordan and Bill. I posted the link twice already, and invited them personally, for comments, corrections, suggestions. Now they woke up. Invitation number three, do I mind? And their comments seem to be pretty good, except that they contradict all that Prasanna was saying. So I told them, but anyway. We’ll find a common base, these guys are very reasonable.

Then there was another, “big” meeting, where Eric announced that the entity I mentioned does not exist. I pointed at the line of the code where the name of the entity is present. In short, the guys are in denial.

Meanwhile, David decided to teach me some software development. Historically, he wrote, writing unit tests was not the developers responsibility. Sure, in 1992. I quoted the wikipedia. He wrote “I don’t care about what wikipedia says”. Meaning, basically, that there’s no point in discussing with him anything. He just has an opinion. Me too, now what. He also noted that I used their library, so it’s my responsibility to write tests for their library. Etc. I was thinking he’s a sane reasonable man.

The problem is that Prasanna is on vacation, so anybody can say anything; if Prasanna were here, he would slap their asses and introduce some common sense.

Meanwhile, seems like I’m getting closer to the root cause of the failure. Copy-pasted more of their code into my server; blocked any usage of the server except on dev machine, pushed, asked Stephen to review; so he did. And I merged.

But I could not build: the build was broken. What, where? David did it. Ok, not having time for discussions (it was late), I fixed one of his compilation errors (I thought that was the only one); ran tests,

pushed, merged, built: kaboom, two more. David improved spelling in one common constant name. So even my code did not compile, and one of Eric's masterpieces of shit. Ok, I fixed it, ran tests, pushed, merged, deployed. Good. Ran my tests, got diagnostics, pulled the log for tomorrow; time to bed.

Wrote to Vignesh about the quality of development process, wrote to Gavin asking if we could block a merge until tests pass.

Good night.

July 12, 2022

Crazy day today.

It started ok; the 1pm meeting started with Stephen (Stiffen) showing pictures from Webb and talking about them. Fuck, are not we intelligent?

But... ok, I was still investigating that access permissions failure, when I saw that Eric updated the case I assigned to him. He closed the case, assigned it to me, and wrote that there is a wrapApp method (he wrote it about a month ago) that I should be using - look for samples in other code. No tests, no documentation.

Ok, it did not work either, but that's not what he was interested in. He asked if I run it in my personal dev env. That was known to be impossible for about half a year now. Nobody could reproduce the steps, loosely written in one of four docs (Eric has one favorite, but there are three others); the doc looks more like an opinion exchange ("Bill also said that he tried this, and maybe it would work" - that kind of document) - alchemy, in short.

So Eric tried to insist on me first running "individual dev env", and if I have questions, I should ask server people. Ok, I asked him, is he willing to help me get through the steps if nobody on #server answers? Silence. 10 minutes later I asked him again. Silence.

Later in the day Vignesh wrote to me, Bill, Stephen, that Eric had posted a very good writeup on our slack channel (you already guessed that all the knowledge is transferred via rumors on slack); and then I went there to see what he wrote, and he mostly tried to convince the three of us that we saw Sebastian's presentation from 6 months ago, but never followed. Then everyone got kind of crazy; Bill wrote that Sebastian never demonstrated us how to do it all, he only showed how things are good on his computer, but how was it, remained a secret; then Bill wrote a bunch of hints regarding how to get through several steps (never bothered to document it in the document); but things just don't work.

So, beautiful, Prasanna absent, and we are... let's load it all on Vignesh.

That's not all. Suddenly, Sebastian (I would not qualify him as a programmer, a copy-paster maybe) asked publicly, why there is such a stupid pr, consisting of three commits, no impact on production, just tests, and it has no use in our product.

Ok, I was pissed off for about 30 seconds. Then started writing to him (all in public channel). Something like: “Thank you Sebastian for your interest in the code I wrote. I’m ready to give you explanations on how this thing works and what it is for, and there is a document regarding the structure of the application for which these tests are... and regarding squashing commits, why exactly are you interested in commits being squashed?”

In short, Sebastian, being a simple Chinese engineer, bought this sarcasm. Jordan joined, and we had a nice discussion about squashing commits (why the fuck does it interest them, beats me), but agreed eventually on some common solution. It worked, nice.

So, can we do something productive? We don’t have time for it.

July 13, 2022

Today was a little bit unusual day.

First, Vignesh suggested that we together work on the script for setting up “individual dev env” (not local, in aws), and get it to the quality that he himself could use it successfully. Good point. So far stuff does not work for me, or for anyone else. Eric is silent.

Second, I’m getting deeper into their shit about permissions, and have a feeling that I may just bypass their shit, and exchange with the database directly, controlling the access by myself. Been there, at Google. Eric’s crazy obscure code, we don’t need it (and nobody understand that undocumented bs anyway).

Third, I eventually copied my precious dev scripts to a folder where I put them nicely, wrote clear comments, and will make them work for everyone. Had to give them long stupid names, because people don’t know what does it mean, >c (for us, old farts, it means “compile”, that’s from IBM days). Etc. >nbp means “new branch, compile, run tests, push”. So I also had to write a bunch of aliases.

Will also add scripts to grab logs from dev machine. The best way would be to extract slices from the logs, but that’s next.

That’s it for today.

Pretty tired.

Also, by replacing a bunch of `if(this || that && anotherthing) throwServerError("big explanation")` with `Require(!this && !(that && anotherthing), "big explanation")` already found one bug in Stephen’s code, yesterday, and another in mine, today. The shorter the code, the more obvious are the mistakes.

July 14, 2022

This morning I was planning to go ahead and dive deeper into that fucking bug of Eric, but Santhosh wrote about a new bug. The bug, as it turned out, was this. There was a precondition for updating a record type called `BusinessUnit`, it looked reasonable, but since our precondition checking is still half-way between being imperative to being declarative, that specific precondition just was not being checked.

During the transformation this precondition got into being properly checked. My tests passed, I merged it, deployed - and kaboom, Santhosh sees an error. How come my end-to-end tests don't see it?

First, it turned out that the precondition was absolutely stupidly wrong. So I had to a) fix the precondition, and when I fixed it, some of my tests started failing, so I had to tweak it again. But what about end-to-end?

See what happened. I had that test case - but it was running on a popular instance of `BusinessUnit`, and Santhosh with Jordan asked me not to do updates on that instance. How come I did not create a test case on my test instances? FIIK. It was in Arie's days. Rush, rush, don't sleep on the meetings... etc. In short, everybody's guilty except me.

So I fixed that by lunch (that's 9AM in California), but then I still had to tweak the tools for investigation, in short, I could not connect with Stephen before lunch regarding trying this shit together.

Meanwhile, Bill showed some activity regarding running the server in "personal env"; then he helped me to figure out wtf is in that script of Eric. Of course the whole complexity is accidental. That's what Bill called "devops programming". Devops don't report errors. They write code for a happy path, and then spend time investigating wtf happened. In my case, "SESSIONSECRET" was too old. As a result one server was returning a json with error; the script did not care, it "extracted" a url of some service; the url was returned as an empty string, so further all those curl calls were calling what? Just nothing. `http : /someservice?someparameter=someshitselect * from something where id=12345`; we have about 20 distinct, pretty sophisticated statements, retrieving this and that, etc, permissions, project information, entity information (yes, the schema is stored in tables in the database. No, not one table. And although we have postgres behind... Well.

But I'm having fun. And I'm not alone in this.

Maybe you think: compare this to the cool stuff Google does. Have you seen the guts of GWS, Google Web Server? I worked with it. It's worse.

July 15, 2022

So, Friday, meeting day, but Prasanna is somewhere out, so Vignesh is in his place, so we had two meetings instead of one. And Vignesh, unlike Prasanna, asks questions and wants to understand. Is not it a pleasure to talk to someone who wants to understand? I know, I know, trusting would be a bad idea, but anyway.

So, I've been working all day on that stupid script. Replacing python calls with jq, reporting errors if there are errors. And the like.

It more or less works right now.

Meanwhile, this morning I merged a pr with one small thing, implemented `toString` in one simple class. Did not even run end-to-end. To be more precise, some people did it before me. NPE in the server. OMFG.

Meanwhile, in the meeting, I suggest Stephen that I help him with a bug he was assigned, and then we'll work on my big stuff. And I started. But while I was just reproducing, Stephen discovered that NPE. He just rolled back that stuff of mine, but well, WTF. This weekend I'll try to reproduce it. Tomorrow morning, for instance. How can it happen, I don't know. And how come error report is so stupid, and how come there's no stacktrace in the log. Will see.

So, what else? We did not have time to go ahead and investigate that shit I've been investigating.

That's it, basically. That's it for today.

July 18, 2022

Well, yesterday I wrote a more or less functioning script, sent it for a review, and upgraded the document. These scripts, no need to deploy them, just merge them.

The only problem was that SESSION_KEY was expiring too fast, at times I couldn't finish running the script. Now Stephen showed me a trick: we need to exclude our company from the Google's list of applications, and it'll work.

Now I can, as Eric taught me, run my stuff on the cloud. Yes, it's slow. But it's faster than on a dev machine.

Aha, and I found the causes of their NPE. They take, like suckers, get a `java.lang.Integer`, and, somewhere deep in the guts, call its `.toString`. Programmers. These people are Scala Java people. Well, whatever. I'll fix it.

And, since the script is pretty slow (files are huge, they are being deployed slowly... I'll fix it later, I guess), so I have time to do French, to watch movies, to write docs.

Ok, we are close.

But omfg, the management. In the morning I happily reported my progress, this and that, etc. In the middle of the day, at the meeting, Vignesh says: ok, since you've finished it, now do this... what a bitch, really. I have not finished, I found a solution. So I told him, it's not done yet. Well, I have to remember, one should not show to an idiot a work that's half-finished.

Whatever.

July 19, 2022

It was a dramatic day. I investigated, wtf `toString` throws an NPE, the source of which we don't see with the naked eye. Got it, eventually. It's Scala 2.11, where an implicit conversion of `java.util.Long` into `Long` may cause NPE, if the source is `null`. Scala 2.12 has it fixed, but we have Spark, which is relatively retarded, mentally, in the sense of Scala. May years ago they were great and bright Berkeley students... well... it's all in the past.

I wrote a fix, sent it for review, my fresh scripts and this fix; sent it also to David, who's kind of playing a role of some kind of expert.

David was insulted, a lot. Hey, why says, why are you, guys, fixing "our" code? ("Our" means something produced by the core team: they think of themselves as the smartest ones; pity they don't know much of Scala.)

Also, he did not like my code style; and he found my scripts stupid, that's what he literally wrote. In these stupid scripts I had just replaced stupid python calls with jq, and stupid scp calls with rsync.

I answered: ok, I got it regarding `toString`, I can handle it without your cooperation; thanked him for one useful comment; but in a personal slack message wrote that if he ever again calls my code stupid, I'll complain to HR, and he shouldn't forget that my experience with Scala is bigger than his, and my knowledge is deeper (because fuck it).

But this idiot continued swearing, so I wrote to Cathy in HR - but did not send it yet.

And now Prasanna writes to me, on Slack. Well, I'm thinking, wtf, it's a good birthday present (yes, guys, it's my birthday today).

For some reason, we got a full understanding with Prasanna, pretty fast. He asked me about the status of the problem on which I've been working for about a month; I gave him all the details, he gave me an advice (didn't work), and explained to me the technical factors that I wasn't aware of (this helps me a lot with my investigation). Ok, here we go. See, we can talk business with Prasanna (that's what I thought). I had kicked out that pull request, and I removed the part that didn't touch scripts, and sent it to Stephen for an approval; he approved it, I merged it.

This bs that "belongs" to the core team, I won't touch it. BTW, regarding that NPE, I wrote a fix and a test, and David's reaction to that test was: "wtf". Is he a programmer, or is he not? There was no test,

now there is. Well, anyways, his opinion is irrelevant. Same as he's not interested in the definition of unitests in Wikipedia. We'll get over it. We'll bypass it (will we?)

But I wrote to Stephen, to discuss all this. What does he think about this "core team" being territorial? Well, Stephen doesn't believe in these "territorial" ideas, and he did have troubles with David, they even screamed at each other. (I never screamed at anybody, I think... did I?)

I won't scream. It's a zoo. Will you, coming to a zoo, scream at the monkeys? You'll just smile. Here are the animals. They show their teeth. Maybe they are smiling at me. I'll smile back.

So, okay, who cares. Okay.

But now that the 'industrial cloud' started working, everything's fine: no need to merge to dev; and I can tweak their "core-team" code, investigating their bug. Prasanna asked me whether it's a bug or a feature (if it's a feature, we have to figure out how to use it). I wrote to him that I consider it a feature, an undocumented one. Yes, it's a month-old, it popped up right before I started investigating it; I'll finish investigating and will report, when done.

So, here we go. So far so good. Not fired yet.

July 20, 2022

Now that I have all the tools I needed, just no support from the core team, I'm working on my own, moving fast.

There was a meeting at 1pm. Vignesh did not seem to understand what I mean (and definitely he does not remember what are these jira items about), he suggested to close the case, because, you know, all the steps I listed there are done. Okay, it does not work, but it's a different jira story. The task is finished, right?

I willingly agreed.

So, the case is closed, now I'm working on another case: make it fucking work.

Prasanna gave me an important hint yesterday: permissions on an entity don't come from a database record. Cool. Now I see where not to look - and investigating their weird, numerous, joined sql statements might have taken another week. But since it's calculated (in an obscure way) in the code, it's easier. More, since I run it on my individual machine, I'm free to tweak the code and add any logging. My future plan is to start using remote debugging, but I'm not there yet. Now I just run a "redeploy" script, and it does everything, and when done, starts `tail -f serverlog`, so I see it and run tests.

The server side does not know yet that it's in a "personal environment", but that's next. The test side does know it.

July 21, 2022

Meetings, back to back, and in the end they ask when it will be ready.

Ok, in the morning had rather philosophical discussions with Bill and Stephen: what should we do if we have a compound key as an id, and at times a component is just missing, and we have no clue what to look for. All these problems were solved over 40 years ago, but not everyone reads book. E.g. Codd (the book is about 50 years old now). Well, whatever, who cares. We just had a philosophy discussion.

Then I found, eventually, the code that is providing these permissions. Lots of sex with this code... it's already 9pm, and I'm still busy.

The scrum was as usual.

Then we had a 1-1 with Vignesh. He told me he's not happy that I complained to HR regarding David, not to him. Sure; the HR informed the manager: see, your employees are complaining! That's it. Also, Vignesh gave me a wise advice: Not to talk to David, and in case of problems just send the compromising evidences right to Vignesh. I guess he's recruiting me as his agent. Also, he was telling me to stop doing all that side stuff, not the Main Problem. Well, he doesn't get it. Probably if I only write the main, happy path, not touching errors handling or tests, he'll be happy. Of course it's a pleasure talking to him when you figure out that you are talking to an idiot. BTW, he's also talking to me as to an idiot. Not that I care, but interesting. He suggested to have a meeting, three of us, me, him, and Prasanna, and I'd write to them, what's up with what I do.

I wrote to them: these are the problems with permissions that I'm working on; added a curl command and what I get ("missing permissions"). Prasanna made an immediate deduction: "you have a wrong code". Wow, here we go. We talked with him about it yesterday, and he forgot it already. Wrote to them in the end of the day that the Code was written by Sebastian, and I'd be glad to get help from Sebastian.

Actually, the test coverage is pretty decent, can't complain. But the code is a kind of hallucination... but what can we expect from Sebastian? And besides, should I go ahead and explain all this to somebody around? It would mean just adding more complaints about the guy, and about myself too. Well, who cares, right? If they kick me out, I'll find the next job.

That's the company style we have here.

July 22, 2022

So, this is the result of my complaints. They've just kicked me out. Well, fuck them. It's only a pity that I worked too much for them. That was a mistake.

But at least, thank God, I did not have time to file a fix for the problem over which I was working for about a month. This morning I found the cause, and a solution. I would not qualify it as a waste of

time.

In short, Vignesh suddenly calls me on slack. Ok, ok, got it. At noon I have a sudden 1-1 with the manager.

He tells me: “We have decided...” And I answer: “Ok, send me an email”. And I close my work computer, for good. Bye.

P.S. This evening I opened my work computer. It’s still not bricked. So I had time to zip my git scripts, and sent them to myself, over my Linkedin account. Because they had closed my Google account, but not my social networks. I don’t need anything else from there.

Also, I renamed my Linkedin account; now I’m Juan Gandhi there, a free-range chicken.