

Kamil's Meeting Notes

Thu, Jun 18, 2025 7:06AM • 34:44

SPEAKERS

Kamil Aliyev, P9

03:08

Recording

Kamil Aliyev 03:11

You Hello, hello, hey, thanks for taking your Time. How's everything? Are you doing? Well,

P9 04:00

yeah, okay,

04:01

Kamil Aliyev 04:06

to help you. So do you remember the one the documents that I sent you, the consent form, and so just a quick reminder, I'm doing my master thesis for University of Tata, and it's focused on software quality, how it's managed in financial sectors and etc. Basically we are doing the interviews with the people who worked in financial industry and worked with the financial systems that can help us so and the singles that you need to be 18, you need to have at least two years of experience. Is that good enough for you?

P9 05:10

Yeah, we actually talked about it. And yeah, if you 10, is this requirements, and it's good. Let's go.

Kamil Aliyev 05:26

So let's start with the planning side. How is software quality of projects planned In early design and planning phase.

P9 05:43

Yeah, we actually talk about quality from beginning, but during the design quiz, we already discuss things like model accuracy targets and what kind of logging we'll need. So let me roll back a bit. I'm a quant engineer, so I can basically talk about only the quant and maybe front office stuff. So we design quality in the beginning, basically of models, and especially in front of his job, pricing, accuracy and latency and other things, we can't leave it to the end, because The goals should be in the project plan.

Kamil Aliyev 07:08

What priority does software quality have in project planning and requirements gathering?

P9 07:30

How you define the quality goals of project as I said, in a model design and development or front office job crucial to basically absolute work quality as a priority. And the reason for that is that, basically it's one of the fields that you don't have opportunity to make the mistakes and the mistakes that you have in in the basically model will cost you a lot. That's why the Quality comes first, and it has the huge priority in project planning and requirements, especially in the requirements. It's basically built into the requirements, the quality part. You just need to implement it while developing the model, and it's implemented as a testing like the unit regression test and other kind of testing,

Kamil Aliyev 08:45

yeah, and who is involved in setting those quality goals?

P9 08:57

Yeah, uh, well, it's a mix product owner will give business goals, and they have this and cons translate that into more technical stuff. So there is basically a step where the business gave their ideas goals, and where our developers act like a translator, where they basically document it as more technically. So if we say, if they say, fast pacing, for example, fast pricing. So we take, okay, like five millisecond is max latency under peak load. Architects sometimes step in when you develop and they set rules for logging or data retention or things so uh,

Kamil Aliyev 10:41

okay, do you use any specific methodologies or frameworks from the beginning of the project?

P9 10:46

I uh, yeah, kind of, if we will talk about the management part way, the project management we use, generally, Kanban, sometimes, then some projects, I know that they use Scrum. But as far as I understood, it's not by the book. So these guys have generally the normal two week screens, stand ups, retros, but our teams generally have a loser, uh, frameworks, let's say we prefer to use Kanban. We also have the CICD pipelines running from the start. So even early code, it's the test automatically. So generally, when you double code, you don't have tests for testing your code initially, but you have other unit tests and other regression tests. Let's say you need tests first to run in your CICD pipelines, and when you have the future already rolling to the prod, then you already you will have unit tests. It's one of the requirements.

Kamil Aliyev 12:40

Let's talk about metrics. What functional and non functional metrics do you usually deal with?

P9 12:52

Let me think a bit. So the functional ones, generally is the field related things and the technical ones. But biggest one that comes to my mind is accuracy, like does our output mesh market or validated benchmarks, mostly well data benchmarks, we also track the error rates whether outputs pass the sanity checks or non functional I can say about latency, memory usage, EP usage. Some systems we

use the stress test, they use through basically weird market data to see if pricing correct, if The same physically works or not.

Kamil Aliyev 14:18

Well, that's interesting. Which challenges did you face while implementing these metrics to the project?

P9 14:51

I the function. Yeah, one project had a new product type and the model was just too slow. It's generally happens, depends a lot with models. Generally, models are heavy and they are slow. That's why you need to optimize them and prepare them for better. But it depends on the model as well. A little bit different question to answer. One project at the new product type and the model was just too slow, and business wanted it to be in production by the end of months. But once Carlos simulation took like 40 seconds per trade, we ended up parallelizing it farming. We used farming, but it was still tight. Also looking, gets time to time tricky, when the appliance basically wants everything. But the issue is that is that it affects the performance.

Kamil Aliyev 16:06

So agree, this kind of things happen a lot most probably. Can you tell me what matters most in terms of quality in your company?

P9 16:38

I uh, definitely accuracy and speed, if you're let's say, if your icing engine gives the wrong number, that's the potentially reputational damage or manual loss, and also it affects the generally the desk really wants instant responses, Then you also got logging and trans ability traceability these kind of things happen, especially useful for back offices where there is a post trade period and compliance as questions. So yeah, generally, those three things, I would Say, accuracy, speed and auditability, or traceability

Kamil Aliyev 18:08

so any situation where those priorities clashed with the business needs.

P9 18:18

No all the time. I It happens all the time. Let me explain this. So generally, business wants to go leave. So to go live fast, to go to broad, but we're like, we basically, you have the problem with testing part where you have very tight schedule and you don't have much time to test everything properly, but you need to test everything. Another problem is that, let's say one time we push the new product road on Friday, it was bad idea. Crashed, and basically all the evening we needed to deal with it. And singles are rollback. That's why midweek push are the best. I uh,

Kamil Aliyev 19:46

let's switch to tools. What kind of tools or frameworks do you usually work with?

P9 19:56

Language wise, generally, we use Python and c plus we use Jenkins for builds, Git for versioning, we got A custom dashboard, but except that we obviously use uh that's different tools like Grafana keep on a DL key stack as many we use spinar cube for code quality. So this is one of the things as well. And seems heavily use performance testing tools like forgot the name JUnit I'm

Kamil Aliyev 21:06

any tools that you tried which didn't work?

P9 21:10

Well, yeah, I would say selenium. We tried it to basically for FOB UIs. But tests are general tests. Tests get broke all the time with false failures. I would say that was just too fragile. We dropped it and moved to basically back end testing only. Also we tested a third party profiling tool that didn't really handle our modes, threaded setup. So

Kamil Aliyev 21:57

now about automation? What's the good and bad part of automation?

P9 22:19

Of the Code checks. Well, the good part is definitely speed. You push the code within a minute, you see that pipeline starts building, testing and if anything basically breaks, you see, not immediately, but pretty fast, we run the unit tests, regression tests and some performance tests. But the bad side is that people start relying on this kind of things too much, they see all green check marks. And I assume that code is great time to time. It's not the case, basically, but if the tests are weak, they miss the real problems. I would say that's the bad side. I

Kamil Aliyev 23:19

did that ever happen to your team?

P9 23:33

Yeah, once we had the bug in edge case, which basically wake up all the test screen. But it don't mean that the future is working properly. Just means that the existing functionality works properly. Well, we didn't test it. That edge case, that was the case. Ice broke production. I didn't broke a production, but we got effect, which we needed to fix, and we added a lot of the Saturday. I

Kamil Aliyev 25:10

Have you ever had a delayed feature just because of the course checks i

P9 25:25

Uh yes, definitely, yes. Once we had the volatility model, it worked, but tests were lucky, and basically we couldn't tell if they will that it was called or the test framework, obviously two weeks to start out and fix it, and basically You have the problems to get approvals when you're half have testing problems, I would say that it really The value in it take that week.

Kamil Aliyev 26:17

So it's interesting, how you handle prioritization when there is a pressure on delivering first.

P9 26:49

I it's like we talk to business, and sometimes we give them an MVP that's safe. I generally team can push things down with correct argumentation. If you had many arguments and sometimes rules for release because you lack of testing, or generally because you have the problems with testing or lack of testing, people will not sign it if they are not sure that all the edge cases and other things tested, still, Sometimes the situation happens. I let's talk about compliance.

Kamil Aliyev 28:47

How do you make sure to meet a regulatory requirements? And what kind of regulatory requirements are there?

P9 28:57

I we have set this for every release, things like logs, audit trails, encryption, other kind of things we have PR reviews, also model changes need to be reviewed. Kind of or anything used in pricing or risk goes through validation and a chain of science, compliance, legal and other internal teams need to Check everything, sometimes even the external auditors are involved. You

Kamil Aliyev 29:47

it sounds heavy. Any challenges with that, like with the complex checks,

P9 30:18

definitely one time I use, I tried to use some new library for curves, those open source MIT license, I would say total safe, but kam plus five did, because it wasn't in the approved list. We have to get it reviewed and then it delete us a little while. So now we can check first. We always check first,

Kamil Aliyev 30:55

does this have a limited innovation? I And

P9 31:04

yeah, generally it does. It's highly regulated industry. I'd love to use new tools like new IT tools. There are also a bunch of awesome tools there, but if it's not on the list, forget it simply, so you end up reinventing some things sometimes or using slower methods just because it's already approved. Yeah, it happens. What about the culture around the quality, like, in your team. Oh, it's definitely good. I wouldn't say that, like AD team is perfect, but people do care, and the team is doing, like, the heavy lifting, especially quantum teams, and nobody wants to build a person who broke the road. Also, there's several sign of chain people in sign of chain requires you to wait and get all the approvals, and they ask their own tests that you need to test before hitting approvals.

Kamil Aliyev 32:52

If you would improve one thing in your current case setup, what it would be

P9 32:57

so let's see my realistic data, the test data is problem, especially with the way the synthetic inputs. Synthetic inputs are clean, super clean, but the real market data is messy. I love to unbox the mimic that's and post mimics production better.

Kamil Aliyev 33:33

Final one, how do you personally feel about software quality? Do you think it's understood well across teams?

P9 33:38

I That's a good one. I think devs understand it. Business. I'm sorry, it depends generally business is already trained well while working with teams, and they understand it. But the features, deadlines, edge cases, production issues, time to time. It's a basically shared journey, but I would say it's understood well, I can't say it's not, well, that's fair.

Kamil Aliyev 34:22

All right, I think that covers everything. Thanks a lot. That was super useful. Thank you. I've developed it

P9 34:36

Thank you.