

# Kamil's Meeting Notes

Thu, Jun 26, 2025 8:49PM • 21:14

## SUMMARY KEYWORDS

software quality, financial sector, design reviews, SRE practices, metrics tracking, resilience, deployment safety, automation, error budgets, compliance, CICD pipelines, incident process, human errors, staging environments, monitoring tools

## SPEAKERS

P10, Kamil Aliyev

00:00

**Kamil Aliyev** 00:54

Um, happy about you to be here. Thanks for joining. Excited to hear about your perspective.

**P10** 01:11

Hey, happy to be here. Always good to talk about reliability and the quality itself. It's a bread and butter.

**Kamil Aliyev** 01:21

Great. I remember you already checked the consent form and the guidelines. But let me start by giving you some information. So I'm basically writing the thesis for master's degree, for my master's in University of Tartu. And the goals for the interview is that, so there is a criteria is to be an 18 and to have at least two years experience in financial sector. And I hope you fit. Can you confirm it, please?

**P10** 02:09

Yep, yep,

**Kamil Aliyev** 02:13

have you do have experience with implementing software quality?

**P10** 02:19

Yes, I do. Okay, great,

**Kamil Aliyev** 02:31

great, great. Let's start simple. How do you deal with quality during planning stage,

**P10** 02:49

we try to get involved early, like during design reviews. Our job is making sure that things just work. But also also work under the stress, so we ask questions about availability, failure scenarios logging as part of the checklist for the from the day one for our team.

**Kamil Aliyev** 03:20

And who usually defines what quality means for a system?

**P10** 03:29

Well, it's shared. Product gives us business expectations like uptime and latency. Then SREs devs, sometimes the risk teams translate that into SLAs and SLOs error budgets and so on. Basically the risk evaluations, and we put card rails around it and decide when the issue is turning to incident when it's not good answer,

**Kamil Aliyev** 04:07

do you follow any framework or formal methods?

**P10** 04:12

We loosely follow SRE practices by Google's playbook, SLI, SLOs, error budgets. Also develops culture like CICD, infrastructures, code, automated testing. It's a mix of formal and practical stuff, but it's loosely followed, not fully so.

**Kamil Aliyev** 04:51

We run into issues, I would say, what kind of metrics do you use to track quality.

**P10** 05:03

Oh, there is a lots of metrics there, like uptime, latency, CPU usage, memory usage, file read and write. Also track Deployment Health, how often rollbacks happen? Meantime, to recover, alert fatigue, believe it or not, but time to time, we need to check if our alert setup is saying we run into issues meeting those metrics, yeah, especially with legacy systems. You know, right now, when we are developing systems, modern software development has this thing called you basically need to define how many years the system is planned to work, the commissioning date. Let's say you need to find another alternative, or do some other options, or revolve the system at that time. In the beginning, some apps weren't built in the with the observability mind, so we are flying half blind, basically, sometimes, and time to time, the business SLAs are just unrealistic, five nines, but no budget for redundancy. Let's say.

**Kamil Aliyev** 06:53

What do you think matters most when it comes to the software quality in your organization? I would say resilience. If something breaks, can it recover fast? That's huge. Also,

**P10** 07:08

Resilience. If something breaks, can it recover fast? That's huge. Also observability—can we see what went wrong? Then there's deployment safety. Like, can we roll back fast if a release goes bad? matters. So the the goal is to, can we see what went wrong? And then there is a deployment safety, like can be rolled back fast if your in this goes bad.

**Kamil Aliyev** 07:32

Ever had business goals go against the quality goals?

**P10** 07:38

Oh, yeah. Business wants new features fast, like yesterday, but we need to check basically, did you test everything? Is there is failover? Is there failover to even need a dashboard for this thing? Sometimes we delay the release to fix basic stuff like logging alerts.

**Kamil Aliyev** 08:11

That's great. What I mean, that's awful. What tools do you rely on?

**P10** 08:19

Generally, we heavily rely on tools like parameters and Grafana for monitoring, Splunk for logs, time to time, we use look as well. We use Ansible for infra, Jenkins for CICD pipelines, and pretty much Kubernetes everywhere, basically everywhere. Modern day development really requires us to use pods rather than the physical servers. And legacy systems, it's very hard to do, especially when they are type decoupled, which makes them very hard to separate the containers.

**Kamil Aliyev** 09:12

Is there any tool that didn't work out?

**P10** 09:19

We tried a commercial alerting tool once it was really good tool. But the problem was it was that it was too noisy and hard to customize, basically with best formative setup, which is perfectly working and really align with the industry itself.

**Kamil Aliyev** 09:48

Okay, what's your take on automation and quality?

**P10** 09:55

Well, automation is everything. We automate, tests, builds, rollbacks, even scaling time to time, but it has to be done right if you automate faster. Problems. You need to design your automation really good. You need to understand what you plan to automate and why you are doing it. You know, there is a joke that developer is the guy that will spend four hours on one hour job just to automate it, but it's a one time task, so it never repeats time to time. These kind of things happen. But for a re perspective, automation is basically the key.

**Kamil Aliyev** 10:52

Have you ever seen a way the automation basically backfired?

**P10** 11:04

Yeah, once we had a broken config basically got deployed. I deploy to plus minus, around 2020, services, I believe, because someone skipped validation, took hours to fix that day, we added better

paychecks and basically the review pipeline, the review of the code changes, PR reviews to the pipeline.

**Kamil Aliyev** 11:49

How does your team balance speed and reliability?

**P10** 11:59

Oh, we use error budgets. Basically, it's like an appetite for hours. We can move fast, but if stuff starts breaking, we slow down the way. It's basically, it's the data driven balance. So if hit that threshold that we already put there appetite, then we basically slow down and make sure that this instant broken is patched properly and will not occur in the future, as a quality ever caused a delay. Sure we've locked release before when the dashboard weren't ready and no one tested the environment, but everyone understand that we'd rather delay than wake up at 3am also today. So it's really key to keep the quality up. And if the quality is not up, you will basically flip back far and you will fix it on on your day off, which is not cool, and most probably it will, it will lead to some incident and can potentially damage the organization.

**Kamil Aliyev** 13:46

We talked about organization. Now, how do you handle the compliance part? Because I know that the finance is highly regulated field. Just wonder how it's done.

**P10** 14:00

So the basically is, the thing is that security and compliance are built into our CI/CD. So there, there is a checks and standards like ISO standards. And banking has also standards by the regulators. Also, we got, let's say, regulatory rules. So that's why security and compliance is basically built in our CI/CD pipelines. We have static analysis, access controls, audit trails, logins, basically, and everything is automated. Before production, there is a checklist that's that's a mandatory, and if you don't cover all the checklist items, then you can't go to prod. So a tough compliance experiences, yeah, once we build the monitoring system using open source tools, but it didn't pass the regular review, regulator review, I would say we had to do to basically redo it, redo it with our own internal tool. Before we went abroad. Yeah, it was those tough times. But yeah, it's fine.

**Kamil Aliyev** 15:48

One more question, do these regulations slow things down?

**P10** 15:54

Yeah, they do. But there are, for a reason, we try to automate as much of it as we can so it's less painful.

**Kamil Aliyev** 16:09

What about incidents? You just mentioned it. What about the incident process? How it's look like?

**P10** 16:18

So it's basically a routine, a everything past mine says there on the people who basically monitor systems support. Since happen, like, somehow you you get the something failing, or someone from teams coming to you and saying, Okay, I can't access this service or that service. You jump in and diagnose, we do live rollbacks comes with change. Sometimes we raise basically, if it's under a lot of pressure, times time it happens, especially end of year, I mean, the fiscal year, where there is, like, a lot of services doing a lot of reporting. Yeah, this time you need to basically scale things up. But generally, this pattern is not repeating. It's once a year or some special, occasional, same and then we have this incident meetings where everybody tries to solve, solve the problem. Also we have the postmortems, where after fixing the problem, most probably the day after or the next day, you have some post mortem meeting where we discuss what was the reason of the outreach, lessons learned and etc, basically what we need to patch

**Kamil Aliyev** 18:17

one more question. Do you have any recurring problems?

**P10** 18:20

Honestly, human errors, mistype things like configs, skip tests time to time it was happening now, we blocked it fully. Nobody can skip tests. At least you need tests with better defaults, more guardrails. Yeah, that's it.

**Kamil Aliyev** 18:52

What's the team's attitude towards software quality?

**P10** 18:56

Mostly good. Everyone wants to have the stable system. No one likes waking up at night or getting these calls, but when we get busy, people do cut corners time to time, so we need to remind remind them that there is a problem with it, but yeah, generally, we are approvers. Most of the things that goes to prod, we basically approve it, and we have good review chain which will make you fix the problems before the base

**Kamil Aliyev** 19:45

let's say, if you had chance, what would you improve in your current quality setup?

**P10** 19:54

I would say better staging environments. It's always a problem. I know in many corporations that are we have the same problem. So there is like synchronization problem with prod and staging areas, which basically makes testing tricky. I love a real replica of prod, but it's very hard to maintain, and makes the things you know working with old data or synthesized data is problem, and basically times, times, let you test everything that you already have in prod while releasing something all the workings in your future. And that's it from my side, thank you so much. Thank you. Have a good day. Bye bye.