Twitter Thread by Anastasiia Kapanina

Anastasiia Kapanina
@a_kapanina

Sooo, some time ago I've promised a thread about my job as a product manager for Kotlin Multipaltform Mobile team. And Friday seems like a good day to do it. Let's roll?

What do I do as a product manager for KMM? Well, same things all pm's do: explore my product, its users and the market around it, work on our strategy and goals, help teams with planning, conduct researches, test hypothesises, and so on, and so on.

Ok, ok, before you get bored to death: there are also things I do not because I'm a pm, but because no one else wants to do it - and this shit needs to be done.

Top-3 things I wish I did not do on this job:
1. Writing documentation
2. Trying to think of a proper name for a product couple of years after it was released (ugh)
3. Fixing some content on our landing page with zero knowledge about web development (actually went pretty well)

Now you can try and guess which pages on https://t.co/ffwIQ8psaY were written by me ■

There are a couple of things that are different about Kotlin Multiplatform as a product though.

First: we are somewhat limited in hard data we get. While mobile apps can collect everything, starting from your location and ending with your soul - I can't even take a quick peek at error messages you get after building your Multiplatform app in Android Studio.

Second: Good product manager always uses the product, right? Guess what it means for Kotlin pm's? Yup, writing code. I really hope it will never see the light of day though...

For me there is also an alternative way to explore the product. I was a QA in Kotlin before all that. So when I'm too tired or too dreamy - I can suddenly find myself testing the shit our of Kotlin Multiplatform.



Let's get back to the horrific injustice of not getting all the data in the world about KMM users. What do we do about it? We get creative!

If there is a thing that I learned from this this job itself, not from the books or trainings, it sure is this: Never. Underestimate. Qualitative. Data.

Interviews, complaints in bugtracker, heated discussions in Kotlinlang slack, StackOverflow questions, tweets - it's all data sources. Want an example?

When me and my mentor @etolstoy just started working on Kotlin Multiplatform product strategy, we did not know anything about our users. Why they adopt it? What value do they see? What are their problems?

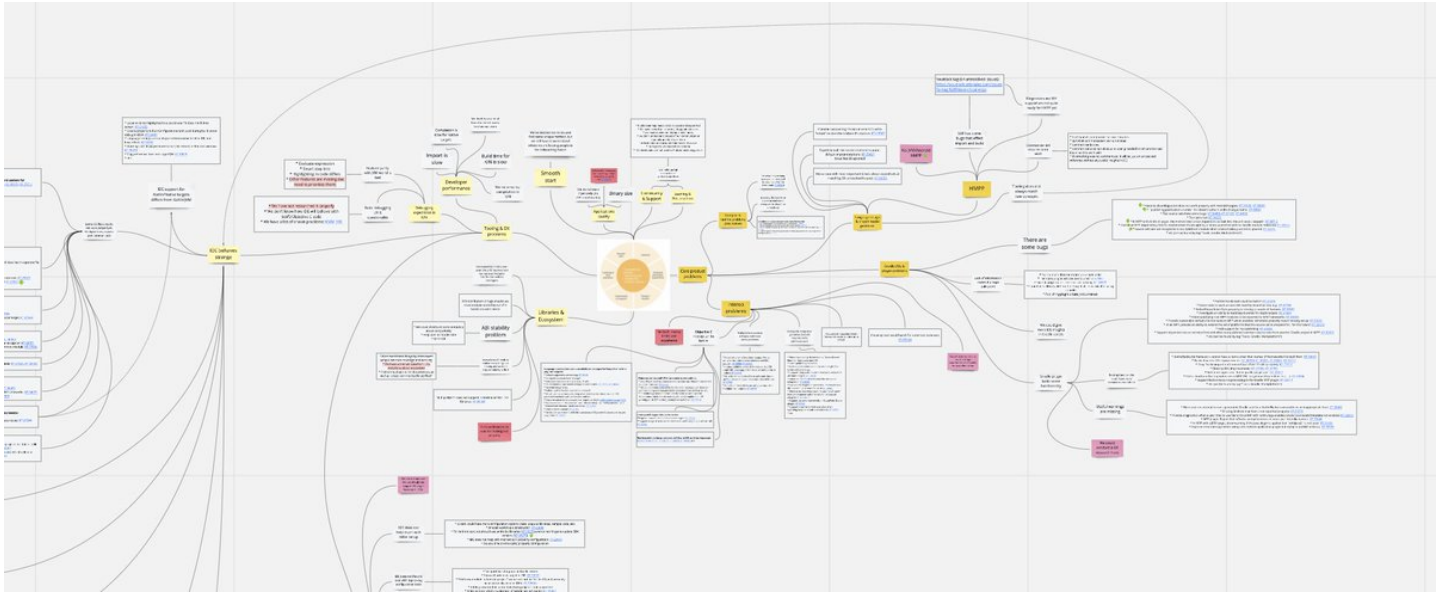
So, we've conducted about 20 or so interviews, not only with Multiplatform adopters, but with adopters of other crossplatform technologies as well. And I also dug through every material on the topic I could find.

Sounds primitive, but we've got about 10-15 major insights just from that. Main of which was our value proposition, btw. It was just sitting there, loud and clear. Not bad for a primitive research, huh?

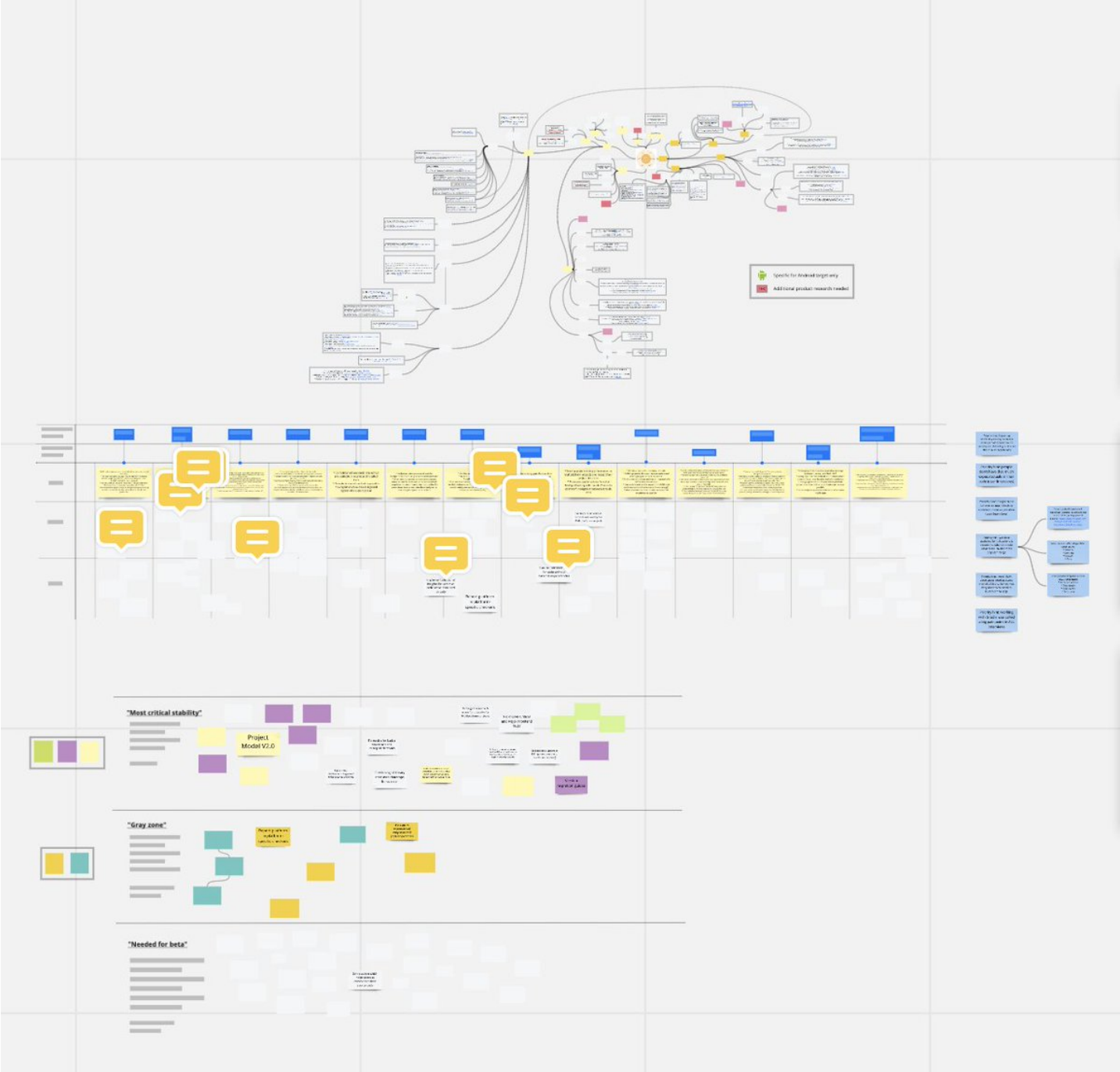
Another example. When we came to the first serious planning in Multiplatform teams, I really wanted to do it right. From the interviews we had general understanding of users' problems, but for proper planning we needed to go deeper.

So, I read every single Youtrack ticket related to us, and then every single slack message in Multiplatform and Kotlin/Native chats for the past six months.

And then I've joined it into a diagram of our problems. This one became somewhat famous thanks to @Touchlab folks.



What they did not know is that this is only the first part of this diagram... ■■■ The second part is our user journey, with ideal picture for each step, current pain points and potential delighters.



And the third one is very high-level prioretisation: we were thinking which parts we need to complete before beta and before stable releases.

Fun fact about this diagram. Recently we've conducted a big user survey, and there was a question "If you'd like to fix ONE problem in Kotlin Multiplatform, what would it be?". It had open answer.

Before processing the answers, I made a guess about most popular clusters. My top-7 was exactly the same as the survey results. We actually know our users pretty well. All thanks to the power of qualitative data!



Not that we're ignoring quantitative data though. We even have a product analyst on the team. He's the dude I always go to when I'm in doubt about how to calculate statistical significance for my findings (in 2021 I promise to get better at that, Max, I seriously promise)

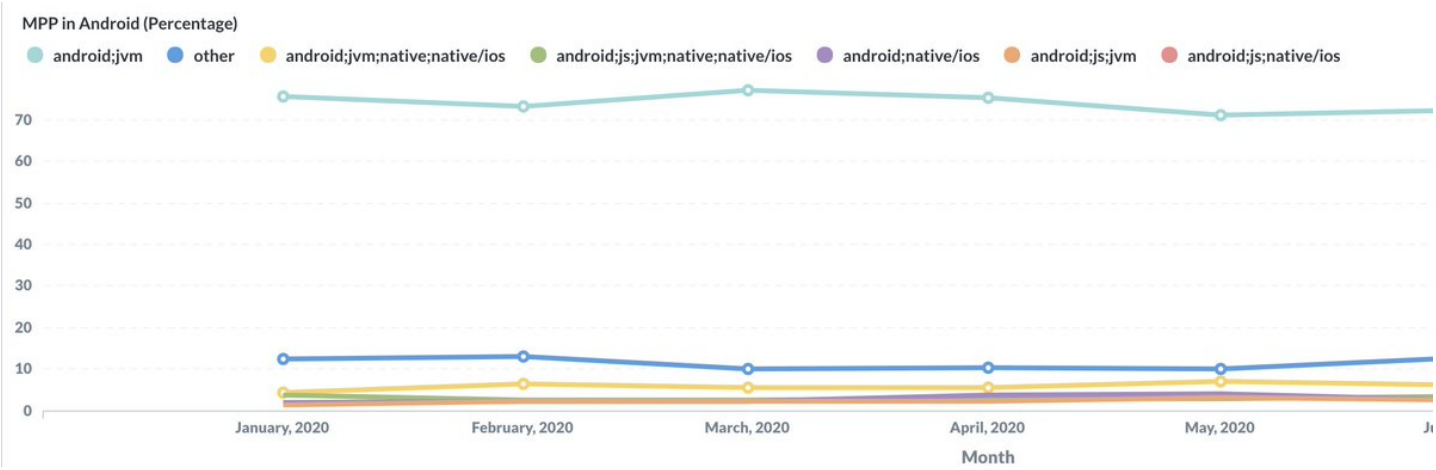
Sooo, quantitative data. For us it's mostly IDE usage statistics (yes, we're limited, but we have some of that), surveys, data from our websites.

CTA: now go and enable stats collection in your IDE, folks. Seriously. I already told you I can't look into your code and laugh at it, what else do you need?

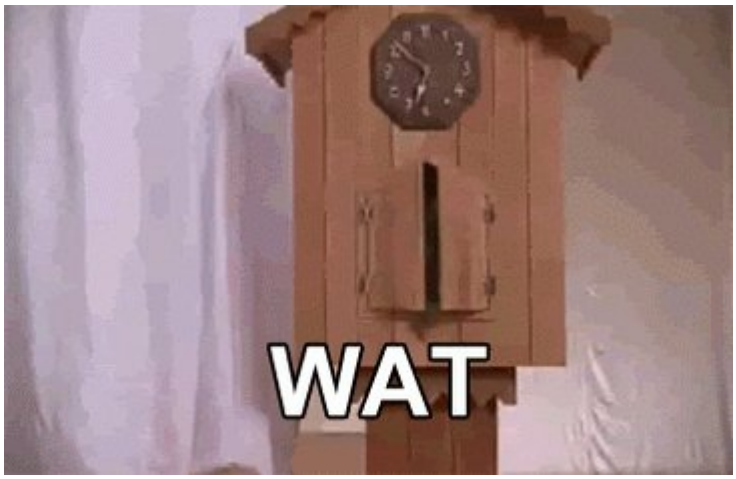
God, those twitter threads are exhaustive. @igrekde, am I even doing it right?

So, funny stories about data from the IDE. At some point we were able to start collecting data about gradle targets that our users build for. This data is collected from IDEA only, per-user, not per-project, on gradle import, not on gradle build. Scared enough?

Not like I was scared when I saw THAT. This is the data for targets that are used together with android target in Multiplatform projects.



https://t.co/f6D5eUIDcx



Combination of android+jvm taking about 70% of the users? Really?! We started investigation. We've discovered that about 1/3 of our users uses jvm target for tests. That our users disable iOS or other Kotlin/Native targets on gradle import. And a bug in data collectors.

I just loved how this one piece of usage statistics disrupted my world for a couple of days - and then put it back carefully right the way it was. Just with new insights.

OK, time to take a break now, let's get some actual work done. We'll continue in a couple of hours.

Let's get back to quantitative data we use to make Kotlin Multiplatform Mobile better. We've talked a bit about IDE usage. There is also a documentation website we collect statistics from.

To be honest, sometimes I use website experiments just to feel like a "normal" product manager.

Kotlin has major releases once in 6 months. Language itself can be changed only in major releases, same usually goes for compilers and runtime behaviour. Even Kotlin/Native started complying to that rule recently! Wish them luck, folks, wish them luck...

Tooling can change in minor releases. Meaning: it can change ~once in a month. For me as a pm that means: waiting for the feature to be released, waiting for people to adopt that release, waiting for them to try the feature out - and only then I can get my goddamn feedback.

It's so much better with documentation! One day I noticed that bounce rate for pages with cases studies was very high: people were looking through the page and leaving. Meanwhile we wanted them to try KMM, not just sit there and read about someone else doing it.

Hypothesis was simple: people are leaving because case studies are poorly linked with the rest of the documentation, they just don't lead anywhere else. We've added a button leading to "Get started" page. In a week bounce rate reduced significantly.

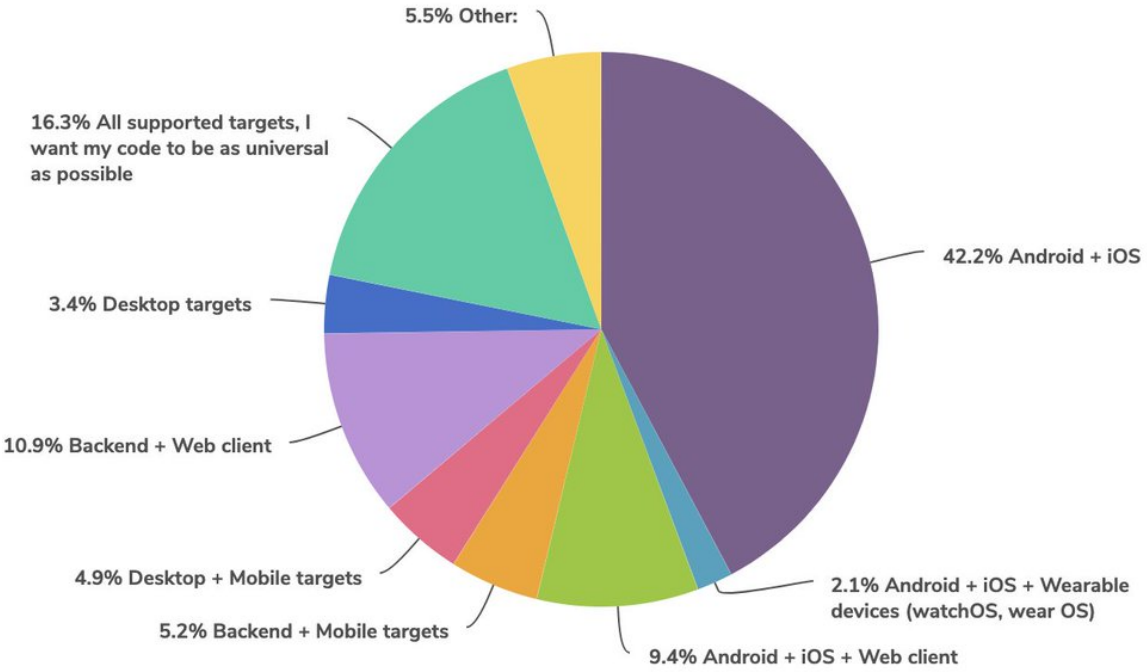
If it was a change in CocoaPods integration, for example, we would learn about how successful it was only in about two months. Ouch.

There are also surveys. I'm becoming a queen of surveys on this job. I did all kinds of surveys, from quick micro-questions in slack channels, just to see some trend - to the big Multiplatform User Survey we've conducted recently, somewhat inspired by DevEcosystem survey.

I'm preparing a post about findings from this big survey right now, but let's have a couple of spoilers here. Folks, who have passed it and know the questions: for what question you'd like to see results the most?

A very basic one, while you're thinking. Multiplatform projects types distribution. Keep in mind that this data can be trusted with 5% margin of error, so we can't say for sure that Backend+Web is more popular than Android+iOS+Web, for example.

7. What platforms is your Kotlin Multiplatform project for? If you have several Multiplatform projects, please choose the one you work on the most or that you consider to be the most representative.



If you don't understand why my whole team is fussing about such primitive thing: we did not have quantitative data for this question before, only the intuition about it. Yeah, as I said, live is hard sometimes.

I would also tell you about UX researches we do, if I myself would be able to finish at least one of them in time. I've started preparing UX research about Multiplatform Gradle DSL, but it's still in its early days.

Main goal of this research is to understand what we can do about very popular Kotlin Multiplatform complaint: "build setup and Gradle suck"

And to be honest, this research scares the shit out of me: it's just so hard to invent the right UX tasks, and I'm afraid not to find anything valuable... But I'll do it next year. Promise.

I guess you've had some sense of where do we get our data and what do we do with it. Let's talk about something else now.

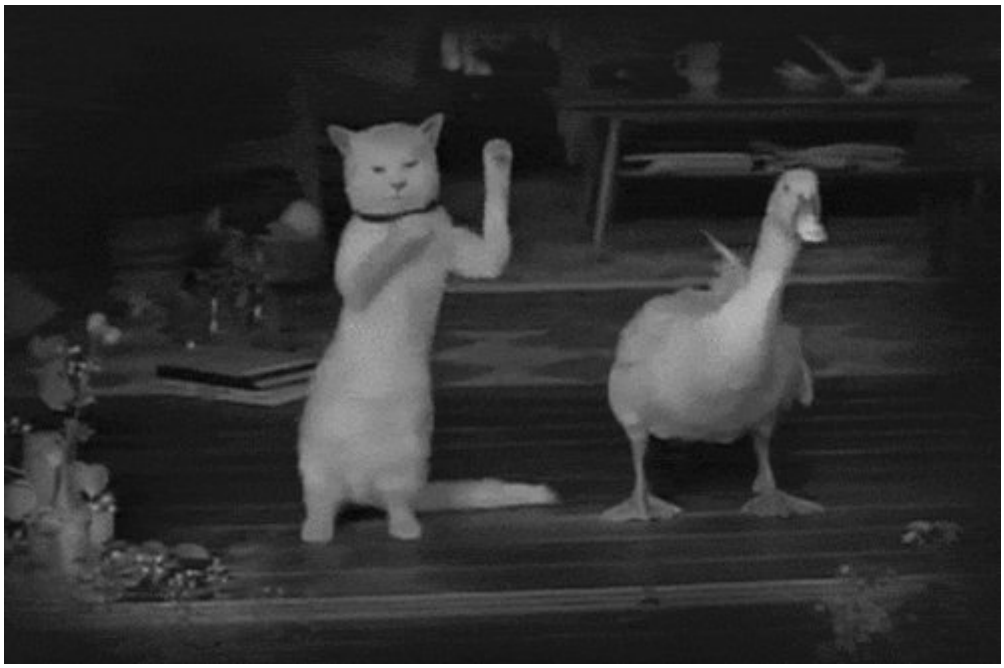
When I first collected ideas for this thread, there was a cool question about what do we do when team members have different intuitions. So, let's talk about that and about avoiding BIAS while relying so much on qualitative data.

When we just started introducing product practices for Multiplatform teams, I was very concerned about people "not having enough context" and constantly tried to educate team members about our users and market.

I was making news digests about mobile development market and about Multiplatform, I shared interview insights, I discussed my researches with them - on both preparation stage and results processing stage.

I'm a lazy piece of shit, so I've stopped doing some of those after a while. But here's what interesting. After I've stopped releasing news digests - my team members started coming to me with interesting news they found themselves.

For example, the article about Netflix adopting KMM was sent to me 7 times. That's when I suddenly realised that my team does have the context they need and that they try to keep that context. Yaaaay!



We also have insiders in our team. @KathrinPetrova, our beloved developer ■, is a former iOS developer. This girl sure as hell can advocate for you, folks! And team lead for our KMM team is a former Android developer. If my intuition is different from theirs - it's a bad sign.

So, what do we do, when intuitions or opinions in the team are different? We dig deeper. Well, ok, we're not saints, sometimes we argue and have fierce debates. But for me such opinions difference is always a good reason to get more information about the subject.

That's basically it about dealing with intuitions difference inside our team. Goodness, they really are saints!

As you can guess from tweets above, my universal recipe for avoiding bias while having only qualitative data on my hands is:
* Think again and slowly
* Ask for a second opinion
* Ask for a second opinion
* Ask for a second opinion
* 42

And since I've started to write about such pretty basic things, I guess it's time to finish this thread. Let's finish it with what I find most challenging about my job.

At thirst I thought that Kotlin being so different from other products is what bothers me. But then I realised that all products are different, and every product manager should use his brain to deal with those differences.

That's basically what it takes to do the job. Brains.

What really challenges me and inspires me now is the responsibility I have before our community. Kotlin community is amazing, and I'd like to make my piece of Kotlin worthy of those folks.

So, that's basically it, but I'm open to any additional questions. Stay safe & have fun with Kotlin Multiplatform!

