1:36

Alexey

**This week, we'll talk about software engineering for machine learning. We have a special guest today, Nadia. Nadia is a software engineering PhD student at the Institute for Software Research, which is at Carnegie Mellon University. Welcome.**

1:52

Nadia

Thank you. The Institute has changed a bit. It's now called S3D (Software and Societal Systems Department) at Carnegie Mellon University.

2:08

Alexey

**Okay. The questions for today's interview were prepared by Johanna Bayer. Thanks, Johanna, for your help.**

# Nadia’s background

2:15

Alexey

**Let's start. Before we go into our main topic of software engineering for machine learning, let's start with your background. Can you tell us about your career journey so far?**

2:24

Nadia

Yeah. I have been in the research field for a long time now – about a decade or so. I started off my journey in software engineering when I was doing my Bachelor's and Master's in software engineering. I'm originally from Bangladesh, so I was doing those in Dhaka. After that, I did some industry jobs for a couple of years, then joined academia at the University of Dhaka, again, in the software engineering department.

In 2020, I started doing my PhD here, taking a study leave for my university. I have been working on software engineering for machine learning now. Originally I started researching in 2014, also in software engineering, so I have done research in many different fields of software engineering – testing, design, business development, etc. I have recently started working on software engineering for machine learning in 2020.

# Academic research in software engineering

3:36

Alexey

**When I hear software engineering and academia, for me, these are a bit of two different worlds. Pardon my ignorance – you will probably now tell me how wrong I am. But I'm just curious. I was mostly a practitioner. I started my career as a Java developer, so I've been doing software engineering for some time. The job I was doing was very different from what I learned at university. So I'm wondering – how is it connected? What exactly do researchers research in academia when it comes to software engineering?**

4:14

Nadia

Yes, looking at this from different domains, software engineering is more connected to industry than the other fields that researchers work on. That's my experience. Previously, as well, when I was working in Bangladesh in software engineering, we had industry collaborations. Companies would provide us with the problems that they were having and, being researchers, we would try to solve those problems using our students or resources, and also people from the company would work together with us, doing research. Even after coming here for my PhD, the first work that I did was interview study, which was again connected to industry, trying to understand the problems that people have in the industry and also trying to come up with ideas from them – what they recommend to solve these kinds of issues. For me, I feel that it's really connected.

Maybe there are some differences when we [inaudible], for example, we’re running a lot of theories, maybe, but that's not practically implemented in the industry, which also happens a lot of times. My impression so far has been that if we're working in the field of software engineering research, then we have to think about the industry. If the industry is not being completed by the research that we’re doing, then it doesn't make sense to make a lot of theories and so forth, if they’re not being implemented. That's my point of view.

# Design patterns

5:46

Alexey

**When I think about this, there is a famous book – I don't remember the actual name of it, but people usually refer to it as the Gang of Four book – which is about design patterns. If I remember correctly, this book is a result of research that these four people did. They researched different design patterns in academia and then they published the book. That's probably a good example of research actually being applied to industry.**

6:24

Nadia

It’s called Design Patterns: Elements of Reusable Object-Oriented Software. I have been teaching design patterns back in my university. I taught that three, four times. That book was really good when I was trying to explain design patterns and everything. Yeah, it was a good example. You have to somehow think that “Okay, if you find a problem like this, how do you approach it?” That’s a good thing, being in academia, you can also look at that from that perspective.

# Software engineering for ML systems

6:58

Alexey

**The topic for today's interview is software engineering for machine learning systems. I think this is a relatively new thing. Data science is maybe only 10 years old – or maybe slightly more. At the beginning, I remember that data scientists didn't really care much about software engineering, but then at some point, the industry realized that this is not how things will be done. Then a new role appeared, the machine learning engineer, who needed to actually take care of software engineering for machine learning systems. Can you describe this field for us? What is software engineering for ML systems exactly?**

7:42

Nadia

There are two ways to think about it. One is, as you said, when data science first came to the industry, we really thought about it as a model-centric thing. People would just create a model, train it with data, it will perform really well in terms of accuracy, and then we’re done. Then, as you said, we slowly realized that it's not what we want – we want this to be used by end-users. And if we want that, there has to be a software system, which the machine learning component will be part of, so that the end users can use it. They don't have to train their models in the notebook or anything – it has to be a part of bigger software. It has to have a UI where they will be able to use the predictions or they will be able to fit it to some different components. That's where the software system comes into the picture in machine learning. The machine learning has to be a part of a system.

Then if you look from the other side – in software engineering, the traditional software engineering processes that we have, we are not really flexible enough to include these kinds of machine learning components in the system. Machine learning comes with its own different properties. It's really uncertain – you cannot really think about a timeline ahead about how much time it will take to have certain accuracy for a specific algorithm. There are also different kinds of working patterns that we didn't have before in software engineering. We have to provide the data and the data scientists will explore the data, and then come up with an algorithm, and then we'll have to incorporate this. We didn't really have that.

There are also some parts of monitoring in software engineering, where we don't need to retrain the model again and again when it's in the deployment step. This is also really different in software engineering. This is where we realized that the traditional software engineering processes and the traditional software engineering impressions that we have – it doesn't really apply to machine learning components, so we have to come up with some solutions to bridge the gap between these different components and make it easy to deploy these kinds of machine learning systems. This is where software engineering for machine learning comes in. It's really trying to facilitate how the models can be used as a part of software.

10:12

Alexey

**There's this famous paper from Google, Hidden Technical Debt in Machine Learning Systems, which has a diagram that went quite viral. I even have it in my Twitter header. It has the tiny ML part in the middle and then the rest is software. This paper is a good example of software engineering for ML systems research, right?**

10:40

Nadia

Yeah, that's actually kind of the beginning, when people really started that. [inaudible] this is a really small part, we have to think about the whole system.

10:49

Alexey

**When was it published?**

10:52

Nadia

I think it was 17?

# Problems that people in industry have with software engineering and ML

10:54

Alexey

**17. Okay. So it was long ago, considering how young the field of data science is, right? [Nadia agrees] I see. In your research, I think you mentioned that you needed to do a lot of interviews and you needed to study what kind of problems people in the industry have. I'm wondering what these problems are. Can you tell us about these problems?**

11:27

Nadia

The paper that you just referred to was an interview study with only 45 practitioners in the industry, from small companies and mid-size companies. We really wanted to focus on the mid-size companies because they have the most problems in industry. There have been a lot of complaints about many different things. For example, to give you a few, when people start working on these systems and they build something, they don't really think about the requirements up front. After building these kinds of systems, sometimes it seems that the client doesn't really care about the property that we're trying to build. For example, if you need a system that needs to respond really quickly and we’re really thinking about the accuracy, not the response time, then it will not be useful to the clients. So there was this kind of stuff.

Then there's also problems of unrealistic requirements, where the managers, the clients, even the software engineers, will think about machine learning as a magic box, where they will provide some inputs and it will just provide them with 100% accurate outputs for that, which doesn't happen. That was a problem of unrealistic expectations from different types of people. There were issues of getting data, there were problems of getting good quality data, there were problems of getting access to data, getting access to the domain experts.

There were also issues of integrating the different components together – the machine learning component and software component – and people really complain about whose responsibility is what, which is not really assigned up front, so the data scientist doesn't know that they'll have to deploy the model. Sometimes software engineers will deploy the model and they will complain about the code quality of those models – that they cannot understand the code and they cannot deploy it well enough. Also the data scientists will complain about, “Okay, the data scientist just changed something in the code, and now my code doesn't work.” So there has been really a lot of dissatisfaction between all of these people. These are just a few examples.

# Communication issues and setting requirements

13:36

Alexey

**So friction between people, because data scientists are not trained to productionize their models, I guess. Then there are data issues and then requirement issues. Requirement, data, and people – three main parts, right?**

13:52

Nadia

Yeah, you could say that. There is also the problem of setting expectations, which is also an issue of communication. In this kind of system, it's really important that you have something written. In software engineering, as well, we don't want to document things right. Being software engineers, we don't want to document anything that we have. But when you have a system like this, which is risk prone, and which has different expectations from different sides, it's also a really important part to have proper documentation so that all the people are on the same page.

They have really different vocabularies as well. People talk in different languages, this is a very common issue that we have faced. I can even give you an example of a term – the term “performance”. If you talk about performance with a software engineer, they'll think about response time. And if you talk about performance with a data scientist, they'll think about accuracy. So it's really that, when it comes to performance, both of you are talking about the one that you are concerned with and then the person in front if you understands something completely different. These are all the problems that we face.

# Artifact research in open source products

15:06

Alexey

**Are you focusing on any specific issue from all this in your research? Or are you trying to understand all these issues at the same time?**

15:17

Nadia

That was one part of the research, where I was trying to understand what's happening. Right now, I'm also working on another project, which is also trying to understand but also analyze some artifacts. For this project I'm working on right now, we are trying to gather a data set for the researchers so that they can analyze the artifacts. For example, all we know from the research side comes from people – what people tell us and what problems they have. We have no way to really look into the artifacts ourselves and figure out what's actually happening. What are they actually documenting? What did the code actually look like? What did the collaboration actually look like from the competition and everything? The way that this has been done in the past is, as I said, from interview study surveys.

Sometimes we have industry collaborations where they'll let us do a case study on a product, but this is also not generalizable for all the different industries out there. Right now, what I'm doing is trying to gather a dataset from open source – trying to figure out the machine learning products that exist in open source and analyze those different artifacts to understand what is actually happening in those kinds of products in open source. That's an ongoing project. I’m still working on that and trying to wrap that up. I do have a small dataset, which I'll publish soon. I hope that will be useful. But that's different research that's going on. I also have been working on some… [cross-talk]

16:53

Alexey

**May I ask about this dataset before we go to the other parts of your research? I'm just curious – what exactly is in this dataset? I understand that the goal is to understand how different people collaborate, and you selected open source because it's easy to get data, right? Or did I get it completely wrong?**

17:23

Nadia

I got it to some extent. [chuckles] It's not just about collaboration, but it's also about the product itself. The products that we build in software engineering are slightly different from the products that we have in machine learning. We only have information about these products from people, but what we want is to study it ourselves. We want to look at the project and look at the project code ourselves. We want to look at how people from different backgrounds are working with different parts ourselves. What are the code qualities? That’s one part of it.

For these kinds of artifacts, we either need access to industry products, or we can do the interviews, as I said. That's why we want to collect a dataset from open source and provide it to the researchers about this somewhere, where people can really have their different research questions and try to analyze those through those datasets. Those datasets will have GitHub repos where they have some machine learning products. Again, we're not looking for toy projects, like small notebook ports or something – we are looking at industry-level products. Those products will have some machine learning components so that it can be called a machine learning system. We're trying to find a list of those in open source. I'm working on GitHub right now and trying to figure out what products GitHub has? It's still ongoing. But to give you a brief description, there's not really much out there in open source.

19:05

Alexey

**I was going to ask you about that. In our community, I quite often see questions like, “I want to learn more about machine learning. Are there good open source projects from which I can learn?” Usually, I don't have good examples because everything I've worked on so far was closed code in the industry. The code belongs to the company and then they typically don't release this code. I’m wondering – how many projects have you actually found so far?**

19:38

Nadia

It's a really tiny dataset. 300-ish I’d say.

19:41

Alexey

**Well, that's good. For this question that people ask, 300 are probably enough. For doing research, I don't know. [chuckles]**

19:56

Nadia

Yeah, exactly. I'd say that's something – it's not completely nothing, but it's not a lot if you want to conduct research on that. It was a bit disappointing for me as well. When I started out, I also had the idea in mind of, “Okay, I'll gather a lot of products and it will be really useful for all the researchers.” I have invested a lot of time from PhD life. And it seems that people really think about it as [inaudible] intellectual property. If you just publish software engineering code, it's not a huge issue that, “Okay, everyone can do those kinds of things. That’s engineering,” but when you create a model, when you invest a lot of maybe GPUs, you do a lot of model training, architectural-based, and you have explored a lot to come up with this model. It's really not that common for people to publish those in GitHub. You can see products in Hugging Face and different platforms, but those are just the model course, not the product that is built around that.

If you want a product, I can give you an example – deep fake, the Facebook product that was really controversial in the industry and academia for its ethical issues. But that is the kind of product that we are looking for. It has [inaudible] that people can greatly install in their computer or mobile and use it. It also has a big machine learning chunk in it. We want products like this. It doesn't have to be a big, big machine learning chuck – it can be a small chunk – but it has to be a product that can be used by the end users, not just a research project, or some toy project that people are trying to learn for a tutorial, for example. That’s the kind of product that we were searching for. It's a really small dataset right now.

# Product vs model

21:54

Alexey

**So in order to call it a product, it should be some sort of application or an API, maybe in a mobile application, maybe in a web service, or something like this – something that you can actually interact with and it gives you some predictions. Not just a model, not just a PyTorch model in a Jupyter Notebook, but all the things around that, which make this model useful, right?**

22:23

Nadia

Exactly. Yeah. For being a product, it cannot just be a framework or a library, and not even an API. Because APIs also, we as software engineers or data scientists can use an API to make a product, so it's not a product in itself. It has to be a complete product that can be used by the end user – the product is then maybe using that API and has some other features and functionality that the end users can use. It can be a complete web application, it can be a mobile application, it can be a desktop application – any kind of application – but it has to be something where the end users are able to use it. If you want to have a product that the industry people built, if you want to have something that is comparable to the industry, that's the kind of product that we want to analyze. That's the product that I have been looking for in the open source.

23:15

Alexey

**Maybe in DataTalks.Club we should start a project like that. It should be useful. I was thinking of doing a recommender system, maybe at some point, because we have all the data. For the people who come to watch today's interview, in Eventbrite, which we use for registrations – we actually have the emails. We can create user IDs, and then we have item IDs, which are the events, and based on that, we can probably build a recommender system. So that would be a nice product, I guess. Right?**

23:54

Nadia

Yeah. If the people… [cross-talk]

23:57

Alexey

**If people use it, right? [chuckles]**

24:00

Nadia

If people actually use it, then yes, of course.

# Nadia’s open source product dataset

24:03

Alexey

**So what is in your dataset? As I understood, this is a tabular dataset, where you have links to GitHub, then some code quality characteristics. What else do you have there?**

24:16

Nadia

We actually have six research questions right now, which we are trying to answer from this dataset. The whole dataset just has the repo names, the description, the links, the stars, the contributors – everything that the Git API provides. Right now, we are thinking about providing some ideas of how this can be analyzed. We're doing it ourselves for six research questions. We have sampled this dataset for just 30 projects right now and we're analyzing those products to get the answers of those six research questions. Then we will publish what we found in those 30 projects. The bigger reason is then the 300 projects which Git links and the regular API properties.

25:14

Alexey

**Can you tell us about these six research questions or is it too early?**

25:19

Nadia

Yeah, I can share it. We're trying to think about it from different phases of the product building. One thing that we're thinking about is the development order of it, whether the model is being developed first, or the product is being developed first – what’s the order of the development in these kinds of products? Then, we are also looking at the collaborations for different people, how the contributors of those GitHub products are collaborating with each other, which modules they are working on – are those modules connected with each other? Are they working on the same module file or the software part together? These kinds of things.

26:02

Alexey

**You can see that from the commit history, right? Are you doing this manually or is there some semi-automatic process that does that?**

26:10

Nadia

For this part, it's mostly manual, but we have some small scripts that help us to look at some specific parts. But it has to be confirmed manually. Some things help that, but yeah, mostly manual. There is more, for example about the testing – the model, the system, the data, these kinds of things. Then there's operations – are they collecting telemetry, are they planning for the model’s evolution? There's also a part for responsible AI, if there's any practices of those seen in the repositories. There is a big chunk for understanding the code structure. In the code structure, we have different things like – how are they using the models? Are these libraries? API? Or are they retraining the model themselves? Do they have a pipeline? How is it automated? So these kinds of core structure questions.

27:27

Alexey

**Was there anything interesting that you found by analyzing these 30 products?**

27:33

Nadia

I'll say we're still in progress. Yeah, I'm still working on that. [inaudible]

27:43

Alexey

**I imagine that in any of these six points, especially for how exactly people collaborate, there are many ways in which things might go wrong. Right? I'm not sure how prominent it is in open source software, but in the regular industry, it’s kind of closed doors, so things do go wrong quite often. I'm wondering to what extent you can see all these problems with open source as well.**

28:14

Nadia

Yeah, it's really hard. We can just look into what we already have in the artifact, and then speculate how it has been there. There are some indicators that we look at to understand, “Okay, why is it like this?” But we cannot answer the whys yet. So the whys can only be answered by talking to the contributors there. That can also be another part of the research I'm doing now. I'm not doing that yet. But when I release it, when people are analyzing the results, they might see patterns that they're not familiar with and then they might want to talk to the contributors themselves. I don't know how GitHub feels about me pinging their contributors and trying to talk to them. It might not be a good approach. [chuckles] Let's see. We're not there yet but it might have [cross-talk].

29:04

Alexey

**You’re doing this in open source, then… or are these projects internal?**

29:10

Nadia

This one is in open source.

29:13

Alexey

**Yeah. So then since they’re doing this in open source anyway, they are kind of opening themselves to this possibility when they decide to do it open source. I guess it's fine. I have a few open source projects and I'm actually quite happy when somebody reaches out to me asking about this project. They’re probably also fine with that.**

29:38

Nadia

[inaudible]

# Failure points in machine learning projects

29:42

Alexey

**I know that a lot of machine learning projects fail. I'm wondering – in this dataset, do you have any cases when the project has failed? Or it's mostly successful projects?**

30:00

Nadia

I can answer this from the interview study that I did before. From this dataset, I have seen repos where there were some machine learning parts before, which are not there right now. I don't know if it indicates failure or they just decided to not build it. That's a separate question that we cannot answer from this.

30:20

Alexey

**What counts as a failure, right? If we understand that the project is not going to be successful, and we stop it, is it a failure? Maybe it's not, right?**

30:28

Nadia

Yeah. We cannot answer that from that one. But from the interview study that we did, we have seen a lot of failures because of a lot of different parameters. People don't really set the expectations up front, at the beginning, so the software engineer might have completely different expectations from the machine learning models and the people who are building the machine learning have different expectations on how that model will be used. They don't really understand the business context of the use of the models. That is one issue that I have found, that comes up very often from people.

There is also that unrealistic expectation part where people want 100% accuracy that the model people cannot provide. From the model side, they have faced a lot of problems from data as well. At first, it will seem that they want to build the model, they want to build the product, and then they will try to figure out the data and they will see that they don't have good data to build a model like that. And it's often because of the AI literacy issue as well. The people who are deciding on building the model don't really know what kind of data they need and if they can gather it, or even if they have it. So that's one issue that the data scientists really face. When they are told to build a model, they don't really get the data that they need to build that model. That's one issue.

There is one problem that I found out where, after people build the model, they don't really think about how it's going to be deployed and what the end users need from those products. How will they be able to use that model? I already told you about the problems of accuracy and response time. It doesn't matter if you're providing the model with 100% accuracy, it's not possible – maybe 99% accuracy – it doesn't matter to the end user, if they cannot use it, if the response time is really slow for example or if the UI that you have provided them cannot be used efficiently to use that model. These are all the problems that we don't really think about up front when building a model. We don't think about the product that we're going to provide to the end users at the end of the day. So that's a big issue that I have seen in industry.

32:48

Alexey

**Did you do anything after analyzing all these interviews? You understood there are these issues – data problems, expectation problems, deployment problems. You know that these problems exist. Are there some solutions for that? Did you work on any solution for any of these problems?**

33:10

Nadia

I haven’t but I was talking to the people in the industry, I figured out that some teams are doing better than others. They themselves have applied some practices to fix these kinds of issues. In my paper I also provide some recommendations about what you can do in certain situations. We don't have solutions to all the problems that we have but as some teams are doing better, they have some practices that they have built on and tried to pass that on as well. But we haven't worked on the solutions yet. We just provided some open questions that the researchers can look at right now to fix those.

# Finding solutions to issues using Nadia’s dataset and experience

33:51

Alexey

**I guess most of the solutions are based on setting up some processes. For example, one of the processes is CRISP DM, which is quite convenient, in my opinion. Even though it's like a hundred years old. [chuckles] It's from the 90s, these design patterns that we discussed. But I think this process is quite helpful. If you follow this, then you probably will reduce the chances that your project fails. Did you see anything like that?**

34:22

Nadia

To be honest, the process is the biggest problem that people have right now. CRISP DM and also some of the other machine learning pipelines that we have seen in the research, and also that people follow in the industry, does not really blend well with the software engineering process. It’s like there are two different processes going on. One is for the machine learning pipeline – how would they get the data and the modeling part and then the deployment and everything. And there's another parallel going on – the software engineering process, which is again, maybe the Agile process or some other processes – the iterative process that they're building on the whole product.

What happens is that the whole machine learning process is being seen as one process for one component. We don't really know how this small component can fit in the bigger product and how we can then blend these two processes together to build the whole product thing. It seems currently that the model is a completely different part and the product is a completely different part, and we don't know how to incorporate these two together. All the processes are also like that. As I told you, the software engineer processes, the traditional processes, it's not really flexible to get that process into their system. Even if you think about Agile, which is really flexible, but again, not flexible enough to think about data up front, to think about the exploration parts of the models upfront. So this is really an issue in industry right now, which is why in the paper, I also say that we need to come up with some processes, which we don't have right now as researchers.

We can keep proposing some but we still have to see the feasibility in the industry whether this process can be followed in the industry at all. Currently, people are proposing stuff, but it's not being useful. Again, in industry, people are trying to figure out their own stuff. But we haven't yet reached one process that is really useful for these kinds of products, which we need to build. This is a research problem we have, and we don't have a solution to that yet.

# The problem of siloing data scientists and other structure issues

36:28

Alexey

**It’s interesting that you mentioned that there are two different processes. From what I see, sometimes these two processes just coexist and we just need to figure out – data scientists and engineers work with one process and then software engineers work with another process to somehow try to work in the same team. Then there are some sort of connection points. Maybe there is an API, and the software engineers call this API with the model, they get back predictions, and then they use it. Is this how this problem is typically solved in the industry?**

37:08

Nadia

There have been different structures in terms of how people try to solve it. One is very common – siloing the data scientists completely. [chuckles] They do their own stuff and provide them with either some model code or some APIs that the software engineering people can use. This siloing creates a lot of problems because of [cross-talk]

37:30

Alexey

**How many ways can it go wrong, right?**

37:34

Nadia

Exactly, there's so many ways that it can go wrong. People don't really understand each other in these kinds of systems. It's really hard to make it consistent, even. We have seen team structures and how they're trying to blend these two, from the model and product integration part. There are, again, a lot of problems based on different team structures. We have seen teams where they just share the model code with the software engineers and want them to productionize it – want them to deploy it somehow – this has problems of poor quality, software engineers not understanding the code from the data scientists, not being able to productionize it well, and the data scientists complaining about losing the performance that they had from their code when the software engineers are touching it. That's one team that is failing like that.

There is a different kind of team, whose model has API's – the data scientist team will provide the software engineers an API and the software engineers will use that API to get the prediction and use it. There are also problems dividing these two parts. We have, again, different kinds of expectations from those APIs and how to use it. Also, there's another issue of having to have a background or skills to deploy the API from the data science part. They have to have some engineering skills. If they don't have them, they don't really want to do this kind of stuff. [cross-talk]

39:02

Alexey

**They give it to ML engineers, right?**

39:05

Nadia

Exactly. We have to incorporate some engineers in the team so that the person can help the data scientists deploy this product. This is one structure. There's another structure, which didn't really scale, but was doing kind of well – this was all-in-one. They have one team with the data scientists and software engineers and everyone is working together to build the complete product. We saw that it can be useful in small teams, maybe with 4, 5, 6 people. But if we increase the number, it's not really scalable. That's a problem with this pattern. So there have been a lot of different structures – the industry people are trying out different structures and how they can work. But all the structures so far that I saw have some problems.

Mostly, what we have identified from here is that it's not really on the structure – it’s on some of the artifacts that we don't really care about. For example, as I told you, we really need to figure out the communication part and how these people are communicating with each other. They really speak different languages. They have different vocabularies. So how can we figure that out? How can you solve this issue? For this, some teams are doing well, when they try to have workshops with these different people and try to share the knowledge that they have with each other. That's one possible way – to figure out the communication part. Also, figuring out explicit vocabularies for themselves like, “Okay, we will not use ‘performance’. We will use either ‘accuracy’ or ‘response time’. Let's not use words that are really ambiguous for this domain.”

We have also recommended documentation, as I told you earlier. Documentation is the key in these kinds of products, because you will really have people from different backgrounds trying to work on different stuff. You cannot really communicate with each other all the time. You cannot sit with each other very often. So if we just document everything that we're doing in the software part and the model part, the requirements that we set, the expectations that we have, and if we share those documents, it’s gonna be a bit easier to communicate about that later on. We cannot really play the blame game. We cannot really say “Okay, I wanted this and this person didn't provide me with this API.” It will all be documented and we can look at documentation and say, “Okay, this is what we expected and now we have this.” We can really compare from that.

Another thing we recommended was engineering. Right now, in these processes, we don't really appreciate the engineering efforts much. This kind of product can really benefit from small engineering work. For example, if the data scientists are really struggling with data validation and everything, they can be helped from the engineering side – the tools, some scripts – that can really be beneficial for those kinds of things. Right now, we can see that there's a lot of help from engineering from the MLOps part. When we’re really struggling with how to deploy this kind of model, then came the MLOps.

Now we have many tools, different ways to use this pipeline – we have made this pipeline based on those engineering tools that we have. That's really one thing that we recommended here as well. And the fourth thing was process. As I said, we really have to have a good process so that all these different kinds of components can work together and can be incorporated together, which we don't have yet. This is still an open question – how can we figure out a process for these kinds of applications?

# The importance of documentation and checklists

42:47

Alexey

**There is communication, documentation… I guess maybe documentation is a part of communication. Then there’s engineering and good processes. These four, right? [Nadia agrees] I’m really curious about documentation because I am an engineer – at least in the past, I was – and for me, this was the most difficult part. When I became a data scientist, I still hated documentation. For me, it was always difficult, and I needed to do this after my code is done, the model is done. Then I need to sit down and document everything. Then it's work for like a week, which probably shouldn’t happen, right?**

**I should document everything piece-by-piece, throughout the entire project. So I'm wondering, do you have any suggestions, any processes for documentation – any templates to make life for people like me easier? So I don't have to force myself, procrastinate on that, and so on. Maybe there are some good tips or best practices for that?**

43:58

Nadia

I haven't worked on documentation, but there are papers and research going on concerning documentation itself. People have provided different kinds of templates or checklists for different parts of this kind of application. For example, Model Cards got really famous for documenting the models – how the models are… this came from an ethical perspective, but it provides you with the context of the model, what the context of the data that's being used is, and how this model can be used for different products. So Model Cards is one kind of documentation.

Then we have seen data fact sheets. Researchers, as well, are trying to figure out what can be a good documentation template and people are coming up with different documentation templates, checklists, for different parts of these kinds of products. But I'd say it’s still an ongoing process. We're still trying to figure out what will be the best minimal and optimal way to document stuff. But we don't have it yet. Sorry to say that, but we don't have it yet. But people are trying different things.

45:16

Alexey

**I was hoping for a silver bullet – hoping that you would tell me “Just do this and you will be fine.” And I would do this for the rest of my life. [chuckles]**

45:24

Nadia

[chuckles] Yeah. For machine learning models, Model Cards are really good. You can try using that. We also have another research where we are trying to see how people are using Model Cards and it seems that people are not using it much yet. But I think it can be a potential thing if you start using this kind of documentation.

45:46

Alexey

**Have you heard about Machine Learning Canva? There is this piece of paper for whatever, with different areas. I don't remember what these areas are, but it forces you to think about different aspects of the model before you start working on this model.**

46:06

Nadia

Yeah, that can be a useful thing as well. Yeah, we have been thinking about these kinds of things too. Just maybe they don't provide you with what you need to document everything – just provides you with maybe some hints like, “Okay. Think about this. Think about what we have.” Yeah, this is also a good way to check stuff, “Okay. Check, check, check. I have. I don't have. I have these in this document.” These kinds of things.

46:33

Alexey

**These checklists are usually super useful, especially for data scientists. When we start a project, we often forget about many things – simply creating a page in our documentation solution, for example, Confluence or whatever. Sometimes we just forget about these things.**

**It’s not because I hate going to Confluence, which I don't enjoy. But I just forget about these things. I'm wondering, maybe you can send us a few checklists that you came across that you think are useful and then we can include them in the show notes. I think many listeners will find it quite handy.**

47:14

Nadia

Yeah. I will send you.

# Responsible AI

47:16

Alexey

**Okay. When we were talking about the dataset you were preparing – this dataset with repositories – you were about to start talking about something else and then I interrupted you. I wanted to go back there because I think that you wanted to tell us about the other research direction you have. So maybe we can go back and finish where we started.**

47:43

Nadia

Other projects that I'm working on are mostly from the responsible AI perspective. One thing we’re doing that is ongoing, is trying to figure out what kind of explanations people really want in health care applications. I'm collaborating with Yale University for that. We have people from a medical background, people from social backgrounds, and we're trying to figure out the different kinds of explanations that people want in these kinds of medical systems. We had one use case in hand, which is [cross-talk]

48:20

Alexey

**Sorry, I will interrupt you. I'm just wondering – what do you mean by explanation? Explanation is when we have a model and then the model gives some results? For example, we were talking about health care, “Is this cancer or not?” This is a bad example, I don't like it. But maybe you have a better one. Anyway, there is some prediction based on some input and then we want to understand how exactly the model arrived at this prediction – to this conclusion. This is what you mean by “explanation”?**

48:58

Nadia

Yes. We started off with a different thing in mind. How are people really taking the outputs? How do they think about any concerns about safety, any concerns about trust? These kinds of things. When we were doing that, we figured out that there is a trade-off that people have on accuracy versus explainability. In some medical cases, you don't want to have a wrong answer. But in some cases, you can leave with the wrong answer, as long as it provides you some explanations. There was one application that we were working on. In schools, there is one app for teenagers where they try to think about the smoke scores that the teenagers will have. Are they willing to smoke? Will they smoke in the future? These kinds of things that they're trying to figure out based on some game data. There was one game that [cross-talk]

50:03

Alexey

**Based on what? A game? Like a computer game?**

50:06

Nadia

A game, yeah. There was one game – a mobile game on Android. In the classroom setup, the teacher asked the teenagers to play a game and based on that, they tried to figure out who are the students who are prone to smoking and then provide them with some consequences – provide them with counselors that will help them to figure out how they should not go that route, if [cross-talk]

50:35

Alexey

**I’m really wondering what this game is. Is somebody supposed to smoke in the game? [chuckles] Then if they agree, then they're prone to smoking?**

50:44

Nadia

There's really many different ways that they're figuring this out. There is supposed to be a psychological part, which I do understand well myself. They would provide you with different initiatives to talk to a group of people who you are trying to be friends with. Who were you being friends with? What are you talking about with people? How are you responding to different questions? There’s many small plays throughout the game. It’s a really long game – I think it’s eight chapters or so.

51:18

Alexey

**It's more subtle than what I mentioned.**

51:21

Nadia

It's very subtle. You're trying to build the character. You're trying to figure out your friends. And through all these ways, later on, they'll have some scores based on which you understand that, “Okay, is this person may be prone to smoking in the future?” So this is how they figure it out. The thing that we're trying to do right now – this is really based on some survey data and we were thinking about providing one machine learning component that will predict based on this score.

If we do that, what are the concerns that people have? What are the concerns from the students? From the teachers? From the parents? From the consultants? What are these people thinking about this score? What we figured out is that they will need some kind of explanation. If the machine model provides them with some scores, they'll have to understand how the score was derived, what the students are doing – based on what activities was this score generated?

52:25

Alexey

**If the model predicts that the student smokes, then their parents want to know why. How did you arrive at this conclusion? Why did their son or daughter be asked questions like that?**

52:46

Nadia

[chuckles] They'll be like, “Oh, my God. No, my children will never do that. How are you telling that to me?” You have to have some kind of background explanation that says “Okay, this score is generated because of this, this, and this activity.” If you don't have that, then you might be in trouble. This is where we were trying to figure out what kind of explanations they're looking for – what kind of explanations did the teacher want? What kind of explanations did the students or the parents want? What kind of explanations did the consultants want if they wanted to consult this student? This is ongoing work right now, for which we're also doing some interview studies with these kinds of people.

We're also thinking about our regulatory perspective on that. This kind of medical software is really critical. What kind of regulations would we want in these kinds of software if we want this to get published? That was the main concern when we started with this project – what will be the ideal regulations to provide with these kinds of applications? Explanations are just a part of this whole thing that is going on right now.

54:04

Alexey

**Is it related to all these things we discussed so far in this episode – to problems that people in the industry have and all these things?**

54:16

Nadia

This is slightly rooted in that, not entirely. The relation will be to the responsible AI parts. In the industry, we were trying to figure out whether data scientists and the software developers are really concerned about responsible AI. We really got a negative answer, apart from the big companies, the midsize and the small companies don't really care about responsible AI that much, which was a bit alarming for us. This is why we were trying to go in that direction and see that, “Okay. What are people actually looking for? What do people actually want? How can we make these kinds of industry applications have certain checks for releasing these kinds of applications?” That's why we went in this direction.

Currently, I'm working on another project on responsible AI, trying to figure out the teams – how the teams work together to come up with the responsible AI concerns, how to check these responsible AI concerns. So this is not actually just from the data science point of view. People really think that responsible AI is the duty of the data scientists, “Okay, they will make some model which will be perfect, which will not have any bias, no fairness issues, which will be completely risk-free and everything,” but we don't really think about the product in the end. It's not just the duty of that model, because the model cannot be unsafe. What's unsafe is the product, because the product is providing predictions or the actions that you were being harmed with.

That's why we have to come out from the model-centric perspective – that the model cannot be unbiased. How can we make the product unbiased? What are the other software components that can help the model to come out from this biased view? What is the other team members’ role in this? Right now, the roles of the team in this responsible AI concern is really hazy – no one really knows who wants to do that. No one really wants to do that. They just try to blame the data science part or the model for that. But we were trying to think about what can be the parts of the team members? What can be the discussions of the team members? How we can think of a responsible AI from the early beginning – from the maybe the requirements part where, “Okay, can we have a requirement that would make this product fair?” These kinds of concerns. That's another work I'm doing right now, thinking about the team boundaries for responsible AI practices.

# How data scientists and software engineers can work in an Agile way

56:55

Alexey

**Interesting. I noticed we have a question from Antonis. I know we don't have a lot of time, but maybe you can recommend a resource where there is an answer to this question. The question is, “How would you advise a machine learning engineer or data scientist to participate in a project that runs in an Agile way? Maybe use JIRA? Maybe it's Scrum or Kanban?”**

**Maybe there are some resources that you can recommend that actually talk about exactly which learning engineers or data scientists should participate in traditional software engineering processes?**

57:35

Nadia

There's no one short answer to that.

57:39

Alexey

**I suspected that.**

57:43

Nadia

[chuckles] One thing I can say is, no matter if it's a data scientist or software engineer, they should be involved with the product from the early beginning – from the requirements phase. If you look at my paper, one big issue was that when setting the requirements, you have to have some model people there when coming up with the requirements. If the model person is not there, if the machine learning person is not there, then it's really hard for them to set some realistic requirements for the model part.

The machine learning person should be involved from the early beginning – they should be in the requirements meeting, they should give input on what the goals that they can achieve are, not just from the accuracy perspective, but also from the data perspective. What are the data they want for building this kind of application? If this data is available? If not, then how can we make this model work? What are the key variables from this part?

The requirements, deployment, and integrations come in the late part, but this is really important to start from the really beginning. There are also parts of testing later on. Once the model is evaluated, you also have to test it – whether it's compatible with the system itself. How are you going to integrate that model to the system? How are you going to test it, if you don't have the expectations from the beginning. You have to have those at first and then go to the next part.

I'll say that I never think about the data scientists being completely separated from the whole team – from the JIRA board and everything – they should be in there from the very beginning. They should at least observe everyone and what they're doing – what are the system parts that they're making? How can they understand the context of that and make the model based on that context that's been developed in the product? That's the shortest answer I can say. [chuckles]

59:35

Alexey

**I think you mentioned your paper, which probably has some answers to this question as well – which you should send us too and we will definitely include this in the show notes, in the description. That's all we have time for today. Thanks, Nadia, for joining us today, for sharing all your knowledge and experience with us. And thanks, everyone, for joining us too. I guess that's it.**

60:01

Nadia

Thank you so much for having me. I really liked having this conversation.