EPISODE 917

[INTRODUCTION]

[0:00:00.3] JM: Ethos Life Insurance is a software company that sells life insurance products. Software is reshaping established industries, such as banking, insurance and manufacturing. In these large established industries, incumbents are adopting new technology as fast as they can, but the new technology needs to be integrated with the old technology and the old business processes.

The slow rate of technology adoption by incumbents creates an opportunity for new companies to spring up, who are building entire new companies from scratch with updated software stacks. Insurance is a gigantic market and it's dominated by companies which have been around for 50 to a 100 years. The established players in the insurance industry are trusted brands, but many of them have not significantly updated their technology from legacy techniques of pricing risk.

Risk pricing is really complicated. There's all kinds of variables that need to go into pricing risk. You could build complicated machine learning models. I think it's safe to say that many of these incumbents are not well-equipped to make that technological shift.

Vipul Sharma is the VP of engineering and Lingke Wang is the Co-Founder of Ethos Life Insurance. They joined the show to describe the insurance business, the technical problems and the software stack of a modern insurance company. Ethos has more than 50 employees and it's growing rapidly, so it's a great case study in scaling a modern company in an established market.

[SPONSOR MESSAGE]

[0:01:45.7] JM: E-mail has been around for longer than I've been alive, but there's been surprisingly little innovation in inbox management. SaneBox is a new way of looking at your inbox, that puts features like snoozing and one-click unsubscribe and follow-up reminders as first-class citizens.

If you are overwhelmed by your inbox and you're almost ready to declare e-mail bankruptcy, try out SaneBox. In the onboarding process, SaneBox analyzes your e-mails and helps you sort them into categories. You can get a free 14-day trial and a $25 credit by going to sanebox.com/sed. That's S-A-N-E-B-O-X.com/sed.

These days, I spend more time in my inbox than I do in front of my coding environment. Back when I was programming a lot, I would spend hours configuring my coding environment, because I wanted to maximize productivity. If you spend as much time managing e-mail as I do, it's crazy not to set yourself up for success with your inbox.

Stop the craziness. Get sane with SaneBox. Go to sanebox.com/sed and get a free 14-day trial, as well as a $25 credit. Thank you to SaneBox for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:17.7] JM: Lingke Wang and Vipul Sharma, welcome to Software Engineering Daily.

[0:03:21.2] LW: Thanks so much.

[0:03:22.3] VS: Thanks, Jeff.

[0:03:23.6] JM: In college, I had a strange experience. That was the experience of somebody taking me to dinner and pitching me on life insurance the whole time. This was somebody I had known in high school. He e-mailed me ostensibly just to catch up. We had dinner and then he spent the whole dinner pitching me on life insurance. Why did that happen?

[0:03:50.1] LW: That's a really great question. I had a very similar experience coming out of undergrad. I met a life insurance agent who was an alum of my school. We met at a bar. He actually convinced me to buy a whole life insurance when I was 22-years-old. It's a career and these folks are salespeople. At the end of the day, that's what their job is, to try and find people who are good candidates for life insurance and convince them to buy a policy.

[0:04:26.0] JM: How much money do the salespeople make doing that?

[0:04:28.7] LW: The average statistic that I've seen online is that a typical agent will make about 70% to 90% of the first year premium in commissions. That number is going to vary depending on how many years you've been in the industry, what product you're selling, so on and so forth.

[0:04:50.9] JM: Who buys life insurance?

[0:04:53.3] LW: The typical person that should be buying life insurance in my opinion, are folks who have families, folks who have something to protect, because at the end of the day, the purpose of life insurance is to protect your loved ones financially, if the worst thing happens to you.

Now there are other cases beyond family protection that one might buy life insurance. For example, a business might buy life insurance on a key executive, or a super wealthy person may buy life insurance for estate planning purposes. For the vast majority of Americans and the vast majority of people, the reason to buy life insurance is for family protection.

[0:05:38.1] JM: As I understand, life insurance is priced through these actuarial tables, right? Historically, these are tables that basically predict when people are going to die?

[0:05:49.8] LW: That's correct. That's correct. Actuaries spend a lot of time looking through population, demographics data, so on so forth to create mortality tables of every thousand people, how many people do they think is going to pass away in a certain year. These tables are created based on the population of folks in a life insurer’s historical data.

[0:06:16.5] JM: How good is that data and how accurate is it in making predictions?

[0:06:21.8] LW: That's a really good question. Some carriers have more data than others. Even if the data is good, at the end of the day, a meaningful impact into the performance of that data is how you execute underwriting. Underwriting is basically the evaluation of an individual person against the category of risk that they're supposed to be placed into.

[0:06:46.7] JM: Describe some of the inefficiencies of the life insurance market.

[0:06:50.3] LW: Life insurance has been around for a 150 years. The practices that life insurance companies are using have been very much outdated. For instance, the sales of life insurance, 99% of it goes through a life insurance agent, like the one that you met, like the one that I met. The distribution force is basically human-to-human. There's very little technology that enables life insurance companies to actually provide an end-to-end sale online.

Most life insurance companies have dozens, if not more of different systems that don't talk to each other, things from the underwriting system, things from the policy administration system, things from the sales management system. These systems don't talk to each other and therefore, make it really hard to deliver an end-to-end, seamless online journey for customers.

[0:07:52.4] JM: We'll get into the engineering eventually. I want to make a little bit more context. Lingke, this is your second life insurance company. Why have you started multiple life insurance companies?

[0:08:05.1] LW: That's a really good question. After my experience of buying that permanent life insurance policy back when I was 22, I eventually came to realize that that policy that was sold to me was really not the right fit for my financial situation. At that time when I made that realization, my co-founder, Peter and I decided we will look into this problem.

What essentially we came to realize was that the industry has made life insurance into this very complex morass. We wanted to come in and help bring it back to basically the roots of what life insurance is meant to be; family protection.

We started our first company, Ovid. This is our first year of business school. Essentially, the goal of Ovid was to help folks who have permanent life insurance policies and sell those policies to institutional investors. That was a way to basically solve a symptom of the problem that I had, which was I had a permanent life insurance policy that I could no longer afford to pay and I wanted to figure out a way of recouping some of the money that I paid into it over the past.

Now, what we realized with our first startup in life insurance was that that's a limited market and we're solving a symptom. What we really wanted to do with Ethos was to solve the root cause of the problem, which is helping people get the right coverage in the first place.

[0:09:40.1] JM: I see. The first life insurance business was basically you've got buyer's remorse, you've got this policy that you're locked into and you want to sell that policy, so that essentially, instead of your family potentially getting your life insurance in the case that you pass away, you basically say, “I'm just going to cut my losses and just sell this to somebody else,” so that person on the secondary life insurance market would end up making the money in the event that you passed away.

[0:10:12.9] LW: Correct. Correct. It was a way for folks to liquidate this large asset that they’ve pay a lot of money into.

[0:10:21.2] JM: That's an interesting business to start with, in the sense that it is like a limited market. I can imagine the liquidity being pretty low. I don't know anybody who has – well, I guess, this is – that thing doesn't come up in daily conversation. I don't know who has resold a life insurance policy, or who buys these life insurance policies.

I imagine at the time, you were like, “Maybe we could be a category creation, or maybe we can increase the liquidity of the market through technology, but I can also just see it not working out.” Or you just saying, “Well, let's just go after the main part of the market, like the initial sale.” What did you learn from that first life – what else did you learn, other than the market in a little bit more depth? What else did you learn from that first life insurance company that you're applying to your current one?

[0:11:11.4] LW: Yeah. We learned a whole lot about life insurance in general. When we first started Ovid, Peter and I had zero life insurance experience. There was a massive learning curve in order for us to get up to speed in the market. That taught us a lot of the inefficiencies in the market, that everything is sold through humans. It's all person-to-person relationships. There is very little technology and all the existing technologies really backdated.

Things like underwriting and policy administration are super complicated problems that carriers have solved by bringing on tens of hundreds of people to tackle. That insight was the key insight that led us to eventually start Ethos.

[0:11:59.6] JM: To get into the technology side of things, Vipul, maybe you could start off by just talking about why does building a life insurance company differ from building a traditional SaaS web app that we might think of, or does it differ? Is it just like any typical SaaS web app?

[0:12:21.2] VS: Yeah, that's a great question. Well in principle, it is not. You're trying to build a delightful product, which is extremely efficient for your customers. For different companies, a customer for Amazon might be looking for a product and getting it shipped to their home as fast as possible. For us, the experience is largely the same, which as a person is looking to buy life insurance policy and we want to get the policy in their hand as fast as we can.

The principles and the team structure, or the technology that is required to make that possible is very much the same. The problem we are solving however might differ a lot, which means that every single day when we are solving problems, we are learning more and more about life insurance, and we're finding solutions to bring efficiency into that system.

[0:13:06.9] JM: Describe your stack.

[0:13:08.6] VS: Yeah. We think about a stack in terms of the user journey. When a person is looking for life insurance, the journey starts with, “Okay, what do I do? Where do I go? Where do I buy life insurance from? How much do I need? What am I going to pay?” From that search, they end up finding somebody who is ready to explain to them what life insurance is all about.

Largely, it is an agent. In our case, it's an automated product. It starts with what we call acquisition, then it goes to a channel that we call growth, which is converting people who are looking to find life insurance into a user that will meaningfully engage with our product. The next journey is engagement, which is the application-submission process, which is a very similar experience like TurboTax.

Once the application is submitted, we want to make sure that that policy is activated as quickly as possible, which is the next journey, which includes underwriting and other validation steps, which is what we call activation.

Then the last step which is administration is making sure the users know what their policy is, they can make changes to it. We care about retaining them for the life of the policy. This is essentially the user journey.

The stack that we have is pretty much built according to each part of this journey. Some teams are heavily focused on understanding user behaviors and what they need from qualitative and quantitative perspective, so they are very deep into data. Some teams are working on providing the best experience to users, which means they are very heavily thinking about user interactions, design seamless experience, that part of things.

Some teams are very much thinking about how to bring efficiency into the system, which is making sure the underwriting is happening as fast as possible. We are able to connect with use in back-and-forth conversation as efficiently as possible. That's generally our stack.

On the technology front, we are hosted in AWS. We use cloud for most of our things. We use all the modern tools that are available to us to actually bring this technology, or convert this into a product.

[SPONSOR MESSAGE]

[0:15:22.0] JM: Logi Analytics is an embedded business intelligence tool. It allows you to make dashboards and reports embedded in your application. Create, deploy and constantly improve your analytic applications that engage users and drive revenue. You focus on building the best applications for your users, while Logi gets you there faster and keeps you competitive.

Logi Analytics is used by over 1,800 teams, including Verizon, Cisco, GoDaddy and JPMorgan Chase. Check it out by going to logianalytics.com/datascience. That's logianalytics.com/datascience.

Logi can be used to maintain your brand, while keeping a consistent, familiar and branded user interface, so that your users don't feel they're out of place. It's an embedded analytics tool. You can extend your application with advanced APIs, you can create custom experiences for all your users and you can deliver a platform that's tailored to meet specific customer needs. You can do all that with Logi Analytics.

Logianalytics.com/datascience to find out more. Thank you to Logi Analytics.

[INTERVIEW CONTINUED]

[0:16:45.4] JM: We'll come back to the engineering. I think it's probably worth explaining to the listeners just top-down what happens on the website. I went through the onboarding flow. It was certainly friendlier than interacting with my friend who was trying to sell me life insurance.

Basically, it's a questionnaire. It asked me a set of questions. I go to Ethos Life, I'd say, "Oh, yeah. I'm looking for life insurance." Asked me this curious set of questions, like am I in good health? That one made sense. Then it asked for my annual income, my zip code. I was like, "Why does this matter? Why does my zip code have any bearing on whether I get a cheaper and expensive life insurance policy?"

[0:17:36.4] VS: Yeah, that's a great question. At the end of the day, insurance is about calculating risk attached to the policy. That risk is calculated by several pieces of data. By going through our application process, user is self-reporting a data on themselves. At that rate, data is how healthy they are, which may depend whether what's their lifestyle is all about, what activities they have in their daily life and what profession they are in.

Now zip code is a great example. The demographic within that zip code makes a lot of difference, which means what food you eat, what activities you are in. That actually is a great indicator of your overall lifestyle that you're living in that zip code. All of these pieces of information are discrete data points. When you put all of them together, you're essentially trying to predict how healthy an individual is. Based on that, categorize them into a risk pool, which ultimately decides the pricing for the coverage the user is looking to get.

[0:18:38.9] JM: Could you explain the concept of a risk pool?

[0:18:41.9] LW: Yeah. In life insurance, typically the way you price a product is with what's called a risk class. You might have heard terms like standard, preferred, preferred best, terms like that, those types of terms describe a risk pool that a person might be put into. At Ethos, we have five different risk pools. There can be more depending on the company. What happens is when a person comes in to the application, their application is evaluated against the criteria of the risk pool and they are put into one of those risk pools. That risk pool, it's a major determinant of the price that you'll receive as an end-custend-to-end.

[0:19:33.5] JM: Okay. I think we have a better understanding of the product and we can talk a little bit more about the engineering. I'm getting a picture of this product where basically, I go to this webpage, and so there's this funnel process where you want to get me to fill out a questionnaire. I fill out the questionnaire, it slots me in to a risk pool and then it gives me an offer. I got a couple offers. I got rates for $30 and $50 per month for a 20-year policy. The $50 per year case was $600. It adds up to $600 per year and then that's 20 years and that was a million dollars in coverage.

I was doing the calculation, and so if I pay $600 per year for 20 years, that's $12,000 for a 1 million dollar coverage amount. The implication that I got there was that you, Ethos, you would break even if about one in a 100 people ended up cashing in the policy. Can tell me whether my calculation is correct and just tell me about how you take the inputs and you end up with a calculation, this expected value calculation and the coverage offer that you give to the user?

[0:20:55.2] LW: Yeah, that's a great question. To answer the first part of your question, yes, typically for a term life insurance product, you hope that relatively few of the term life policies result in a claim. That is what the actuarial tables that we talked about earlier summarizes. They summarize what percent of people in a certain pool over a certain time period is expected to make a claim.

Now to answer the second part of your question, how do all of the inputs that we asked for throughout the application process determine a risk class at the end? We asked a ton of different questions, things that you've seen, like do you smoke? What medical conditions might have you had in the past? What are some of the activities and hobbies that you do, so on and so forth? All of that information is then combined with third-party information that we collect on you from other APIs and data sources. All of that go into an underwriting model that then determines what risk class you end up in.

[0:22:04.7] VS: Yeah. You can think of it as a very similar to any machine learning problem, where there's a set of data, there are set of input, there is a model that is predicting something and then there is a validation process of the output of that scope. What we are seeing is that in traditional underwriting, it’s largely a manual process. That's where we see a lot of opportunity in using lot more data and data science, to not only result into better decisions, but also to execute it as fast as possible.

[0:22:33.8] JM: There's so many reasons why this is an insanely good business to get into. I mean, I don't know a whole lot about Warren Buffett's businesses, but I know he likes insurance businesses. I think largely, because it's one of the original recurring revenue sources, right? People don't really turn from an insurance business that assuming they don't get basically taken by some salesperson that convinces them of a sketchy sales pitch.

You need car insurance, if you've got a car. You need home insurance if you've got a home and it's going to be very hard to churn. We know that this is great recurring revenue. Then compared to the competitors, I mean, the competitors are I assume working with just janky PDFs and scanned pages. I'm imagining these dusty insurance offices with lots of printers. Can you just talk about some of the competitive advantages that you can have as a technology company approaching the life insurance business from a total greenfield situation?

[0:23:47.7] VS: Yeah, that's a great question. The opportunity is in every part of this funnel. Just imagine a person trying to buying life insurance, they watch an ad on TV, then go and find an agent, then schedule a call with an agent, maybe on phone, or maybe in person, talk to them, understand life insurance, go through this application process, which is extremely paper-driven and what many times would ask redundant questions and questions that may not even apply to a person.

Then after that, a ton of medical tests and blood tests for all of this data and it is largely 15 to 18-week process currently. We are trying to bring that process down into a couple of days.

As a company, we are extremely motivated to reduce the time from a person starting in more life insurance to the time that they have it there, the policy they have in their hand. What is our leverage? By using technology when somebody looks at our ad on Facebook and Google, they directly jump into an application process. Really thinking deeply about that application process, what questions should be asked based on what we know.

If you're answering certain questions, maybe many other questions don't apply to you and we may not even actually show you those questions to you. Where can we use third-party data to bridge the gap, so that you don't have to go through that lengthy application process?

Removing the amount of time you are spending and filling up their application. Once you submit that application, how can we have the underwriting process, which is the risk calculation that we have talked about, as fast as possible? If there is any validation, or further data needed for the policy to get activated, how can we do it a modern, seamless way, like most users are used to, versus meeting a person, or taking phone calls, lend the phone calls and things like that?

If you look at it, there is an efficiency built into every part of this funnel. By using data, by using technology and great experience, we can significantly reduce the time that is spent to buy life insurance policy and bring efficiency into this entire funnel.

[0:25:59.5] LW: The other thing I would add to that is one of the major advantages that we have as a business is that we've been able to build an end-to-end technology stack for all pieces of the life insurance process. Distribution, which means the customer coming onto our website, filling out an online application that is then represented in structured data, to be sent to underwriting, which we have an engine that evaluates that structured data alongside third-party information that we receive, to policy administration, the delivery of your policy, the creation of your policy, language and document and the automatic managing of billing.

All of these things are typically different systems and legacy systems at an existing insurance company. We've had the advantage of building all of these things from the ground up in-house, so that they're all interconnected, which allows us to deliver a really positive customer experience.

[0:27:06.5] JM: Did you say that the target is two days to offer the customer a life insurance policy?

[0:27:14.8] VS: Yeah. We wanted to be less than that. That's what we are striving for.

[0:27:20.0] JM: Oh, okay. Oh, so the standard is two days. The industry standard today is two days?

[0:27:24.4] VS: No. Industry standard currently is 15 to around 15-week that –

[0:27:28.8] JM: Oh, my God. Why is it so long and why are you only targeting two days? Can't you get this down to a 100 milliseconds?

[0:27:35.9] VS: That's the goal and hopefully, as we build our technology stack, as we understand this problem a lot more, as we collect data, we want it to be as fast as it can be.

[0:27:46.8] LW: There's a tradeoff right, between how fast and how much data you have to evaluate someone, versus what their ultimate outcome is in price. You can always trade off collecting less data and making a more loose judgement and giving someone a higher price to account for that uncertainty.

What we want to do is offer the best of both worlds. What we want to do is we want to collect a lot of data about a person and be able to make a really accurate, personalized decision, which allows us to really understand the risk and provide a good price to the customer. Shifting that frontier is non-trivial.

[0:28:26.4] JM: Fascinating. In that two-day window that you're targeting, I still don't understand why does it have to be two days? Pretty complex data science workflows get done in less than two days. I guess, are there third-party data providers that those are just the longest? I'm wondering what is the longest source of latency in that two-day target? Is it some third-party data provider?

[0:28:53.0] LW: It could be a number of things. Sometimes third-party data providers are slow. That is definitely true. Sometimes we may need to follow-up with an applicant to ask them more information, because something in their application questions and something in the third-party data did not connect, or they say opposite things. We have to figure out why that was the case. Sometimes, it requires a human to take a look at the case to try and understand what that discrepancy is. Sometimes we have data from our third-party vendors that currently is not machine readable.

There are a number of things that contribute to that two days. You're right, in the long-term goal this should happen in under one second. If we're able to truly get all of the information that we need in an API and structured data format and we are able to process that information in an instant. That is the long-term goal.

[0:29:51.3] JM: It reminds me of we did an interview with Checker a while ago. Checker is that API for doing background checks and the user makes a API call and then sometimes the result of that API call involves going to a courthouse to get some information that's only available through going to a courthouse.

You said that there is some data that's not human readable, so the customer makes a request for a life insurance quote and you need to get this – what's an example of non-human readable – I don't want to give away your secret sauce, but what's an example of a data provider you have to get that does not provide you machine readable information?

[0:30:36.4] LW: Scribble notes from a physician, that's an example.

[0:30:39.2] JM: Oh, geez. How do you have access to that information though? If I make a request to our life insurance policy, are medical notes about my physicians visits, are they somehow available to you?

[0:30:51.1] LW: When you submit a life insurance application, this is the case for every single insurance company that at least I know of in the United States, you have to sign a HIPAA form, which allows the life insurance company to obtain some of those records from your medical doctors.

[0:31:08.0] JM: Whoa. Whoa. Man, I bet you can do a better job at consolidating my medical records than I can. My medical records have basically been lost in the sands of time as I move from city to city. It’s a separate issue, but trying to transfer your medical records from one doctor to another is arduous. Yeah, I guess you're not tackling that problem today.

There was one other insurance show we did fairly recently with a company that is improving the insurance brokerage process. I think that was what is called, the insurance brokerage process. Anyway, they had to interact with a bunch of other players in the insurance market. There are these multistage insurance pipeline procedures, but it sounds like you're more on the full-stack side of things. Are there any legacy providers that you have to interact with, like legacy insurance providers?

[0:32:10.2] LW: Absolutely. Absolutely. We partner with a number of insurance giants, including legal and general for instance is one of the carrier partners that we work with. We work with reinsurers as well. Reinsurance companies are insurance companies for insurance companies. We partner with these folks, because they have tons of capital and tons of experience.

When we sell a policy for instance, our policies are not backed by Ethos, because we're a startup, they are backed by a 100-year-old insurance company and reinsurance company, because they've got the longevity and they've got the capital base to ensure that our customers are well-protected for the 20 and 30 years coming forward.

[0:32:54.0] JM: Okay. Let's come back to the engineering side of things. How much data science is going on today? Tell me about your data engineering pipeline, Vipul.

[0:33:08.2] VS: Yeah, absolutely. We use data for many things; improving our product, taking better decisions, understanding customer behaviors and also understanding health of our company. The way we have organized our data is in three different parts; one, that is purely about data engineering, which includes the data pipeline, essentially instrumenting every part of our app to understand how users are using our application, also the data that we get from third parties, as well as all the data that we have in our databases.

Bringing all of those different pieces of data at one place, so that we can do data mining, deeper insights and various other things with that. That's the pure data pipeline and data warehousing part of things.

Then the machine learning part, which is essentially using this data to fundamentally improve the experience for users within the product. There are lots of things that we do. For example, right-sizing is one, which you wouldn't find a traditional insurance company doing it. We care about our customers that they are buying the right-size policy, so that they can afford it for a long period of time.

We use data science to understand the data about a user and suggest them the policy and the coverage that they should buy. That's a pure data science problem. There are many, many examples of such things that we do to not only educate our customers, but also make their experience better.

All of these things, which is the data pipeline, data warehousing and machine learning is what engineering is focused on. Then second big part of this problem is really understanding how our users are using our product, which means deeper understanding of this entire funnel. It's how our user progressing on every single page, every single interaction.

Where are we seeing the biggest friction where we see the biggest drop-offs in those funnels? We run a lot of A/B test experiments to improve the experience of our product. How are those A/B test experiments doing, deeper understanding of those? All of this is what we call product science. Our product team and product managers are deeply involved into this aspect of really deeply understanding a user behavior.

Then comes to the analytics part, which is also the strategy component, which is around pricing, how we are spending – how we are querying our customers, what is the efficiency of each channel, all of those things is a very strategic piece, which is in our business operations team.

We see data not only solving big parts of the product, but to just make everything inside the company and every single decision-making as efficient as we can.

[0:35:56.2] JM: Right. Basically, if we talk about the data side of things, I'm seeing two problems that are related, but probably use fairly different stacks. One problem is the onboarding flow data problem, in the sense that if you're trying to get people to fill out a form and make it through a funnel for purchasing a life insurance policy, and then there's all these things around do you send e-mail follow-ups, do you show Google ads to people? This is one side of the data problem. It's the architecture of your funnel. That's one very interesting problem.

The other very interesting problem is this data engineering machine learning problem, where you've got tons of data, you've got a user with a certain set of parameters and you want to match that user to the right-sized available policy or policies that you can present them with. I'd like to talk about the stack. We can talk about each of those problems independently, but I'm just very curious about the stack, because I've talked to so many companies that are building a “data platform” that involves an ETL process, a data lake, a data warehouse, several different databases, streaming frameworks, a machine learning framework like Tensorflow, a bunch of random scripts written in Python.

You've got all these different things that fit together in a “data platform.” There's not really best practices around this thing, it's people just duct tape and chicken wire these different pipelines together, because that's the state of the art to the extent that I understand it. Tell me about the engineering decisions you've made around building your data platform.

[0:37:55.2] VS: Yeah, exactly. You touched on many great things. These are the problems that almost every company is trying to solve. The tools might be different, but the principle stays the same. For example, we would like to understand how our users are using our product. What pages they are interacting the most with, what buttons they are clicking the most, what part of the experience that they stop, or close the browser and do not want to engage with the product anymore? These are all things that we can capture by instrumenting our application in the right way.

Then the third-party data and the data that's stored in our database. You have to bring all of these three things together, so that you can make deeper insights on it. Because if a piece of data is sitting in one database and another piece of data is sitting in a second database, it's very hard to join across those two things to get deeper insights. That's essentially what the data pipeline and the data warehousing problem is.

What is our stack? We use PostgreSQL for our main database, we rely on a lot of Amazon tools, which are out of the box. Largely, we are still early in our journey as a company, so data volume is a significant problem for us. What we are taking decisions on is the efficiency and the speed of execution. We rely on a lot of third-party tools to bring this data into the data warehouse. We use Redshift.

Once the data is in Redshift, we have an ETL process, which is largely driven by Airflow, where we can actually convert this piece of data and transform it into different tables, or views on which we can drive insights very easily.

Now once these tables and these views are ready, we expose them in radius space. On machine learning applications, could be using this data to generate features, to train the predictive modeling and generate outcomes of it. Or our product science team might be generating funnels on top of it to understand how is a user going from one page to another page and how much time they took and is the performance right and where the drop-offs was.

Our machine learning stack is largely driven by Python. We're still not using deep learning, because we're very – our problems are extremely related to regression problems and problems that are more predictive in nature. We are using things like logistic regressions and few other algorithms, using some really amazing Python toolkit. We drive funnels and charts using a lot of third-party tools such as Heap, Chartio and things like that.

[SPONSOR MESSAGE]

[0:40:31.3] JM: ProsperOps is a platform for automatically managing reserved instances on AWS. Reserved instances are one of the most powerful tools to reduce your cloud bill, where AWS gives you discounts of up to 75% in exchange for a term commitment. You don't have to change your app, or AWS environment to use reserved instances, so your savings start immediately. However, using reserved instances can be both complicated to figure out and time-consuming to manage.

ProsperOps simplifies that complexity by using algorithms to build and manage a reserved instance portfolio that maximizes your savings, even as your EC2 footprint changes. Go to softwareengineeringdaily.com/prosperops and get a free reserved instance savings analysis.

ProsperOps can more than pay for itself with the savings that it can generate. Check out softwareengineeringdaily.com/prosperops and save time and money with ProsperOps reserved instance management. Thank you to ProsperOps for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:41:47.1] JM: As you're starting to get into this machine learning-driven approach, have you noticed any pricing differences between what your machine learning models suggest and what the insurance giants recommend traditionally?

[0:42:08.1] LW: I think we have seen some of that. We're very careful to make judgment calls on that at our current stage, because of the limited amount of data and experience we have. We in conjunction with our insurance partners spent a lot of time looking at not only our data, but also historical data and over a 100 years of experience on life insurance mortality. For sure, but we are still very much in the learning phase and exploration phase of that.

[0:42:38.1] VS: I would reiterate what Lingke just said a little while ago, that pricing and how much data you collect are two things that we have to balance against each other, because it's essentially a risk calculation. If we have users spend a lot of time in our application process, maybe the underwriting process would be faster, which may result into a cheaper price. We might give them an experience, which might not be very efficient, because we are taking them through a lengthy application process. It's a balance that we have to keep in mind to how to make sure the users are deeply engaged with the product, but also get the right price for what their needs are.

[0:43:16.2] JM: Okay, interesting. Let's go a little bit deeper on a specific machine learning model that you might be building. Can you just give me an example of the end-to-end workflow for – maybe you can give me any example model that you want to talk about, just the process of getting the data, maybe cleaning the data, putting the data into database, or data lake, or in-memory system, whatever. Tell me how the sausage is made.

[0:43:47.6] VS: Yeah. I already talked about right-sizing, but I'll choose another problem just to bring diversity into our discussion. Churn. Churn is something that we deeply care about, because we want to make sure that users are with us for the entire lifecycle of their policy. Churn is essentially a predictive modeling problem. When people are submitting an application, when they are getting the policy activated, we want to make sure that everything is going right in order for them to stay along with their policy.

Now if you want to predict the churn, let's say we want to predict whether somebody who is engaged with us and got an activity will churn or not, the data of the applications that they fill would be not PostgreSQL database. We will take that data and the third-party data that we have on them and bring it into our data warehouse.

Now our data warehouse has this entire set of data that based on their application, their behaviors, how they're using the product and all the third party data in one place. Now a machine learning problem is essentially saying, given what we know about this user, what's the probability of this user churning in two months, three months, six months, eight months, 10 months? It's a predictive problem.

We look into areas of these signals that we build just like a regular machine learning models, we come up with initial set of features, or attribute that, we think about are important to make this prediction and then we train it based on the data. For example, a set of people that have not churned and a set of people that have churned, and our machine learning models will now see those two things and start to wait these attributes based on the data they're seeing and will result into some model, which will have weights for each one of these attributes.

Now when a user, a new user comes in, we can take them through the same pipeline where we have all of this data for this new user and we take those attributes, can work the data into those attributes and based on the model that we have built a while ago, we can now predict whether when will this user churn and what would be the key attributes why they would churn?

For example, one of the things that we discovered when we look at churn is that one of the big reason that people churn is that credit card expires and they forget to make payments. Now that by being a technology company is very easy for us to solve, is build the notification systems into the pipeline so that we can now remind users that, “Hey, your credit card is about to expire and you might actually churn and lose the policy. Please go ahead and update it.” By using machine learning, now we have solved a critical part of this churn problem.

[0:46:24.2] JM: Let's zoom in on one particular aspect of this, so the data warehousing. The definition of data warehousing as I understand it, the modern definition is if you want to get a bunch of data into a place where that data can be accessed really, really quickly, you often put it into a data warehouse. A data warehouse is useful for performing a lot of calculations over large data sets. It's expensive, because I guess this data is usually in-memory, or it's in a place where you can access it faster.

You can't just keep all of your data in a data warehouse. Can you tell me about how a data warehouse fits into your engineering process? When do you put data into the warehouse? When do you take it out of the data warehouse and what systems are performing those, I think that's called a ETL, where you're putting data into the data warehouse and taking it out?

[0:47:24.8] VS: Yeah, absolutely. The reason why anybody would use data warehouse has many things to do. One of the things that you touched upon is the efficiency of the query. When I run a query, how fast it can run. Without a data warehouse, if the data is in-memory is one aspect of it, but imagine if your data is stored into three separate databases.

Given every company's stack, would use multiple data stores to store their data. They might be storing FLAC files, or third-party information on S3, they might be storing structured data, relational database, like MySQL or PostgreSQL. They might be using a NoSQL system, where they want to get high through reads output by writing fast, something like Cassandra or MongoDB.

Now the system data is spread into these different locations, where you need a query, where you need to join across these three different things, it's extremely hard because now you have to figure out a way to connect the dots between three different stores. That's what data warehouses becomes really handy, because now you can take all of this data and bring it into one central place.

Now not only you have – you're saving time on network, but also all of this data sits very close to each other. You can structure it in a way whether if it is in different tables, then you can join across those tables. By using modern systems like Presto or BigQuery, you can do it extremely fast, or you can go through an ETL layer and create a view, or a table where you bring the data that is needed for a query into one place, so that you can query on it fast.

Data warehouse not only solves the speed problem by executing a query fast, but also made data extremely accessible by bringing up different pieces of data in one place and making it extremely easy to access that data.

[0:49:15.6] JM: How do machine learning frameworks and streaming frameworks, how did these things interface with a data warehouse, or to what extent do they interface with a data warehouse?

[0:49:30.6] VS: Yeah. In a very simple way, data warehouse is your large database in which you can store your entire data set at different views. For machine learning problems, you need to be able to get that data, run your models, train your models, bring output from this data set. Now some of these problems are batch processes. For example, you don't really need real-time decision-making in order to make decisions.

For example, for our churn, if we run it once a day, that would be fine for us. If there is any other problem, for example if we want to reflect in real-time any part to our customers on our product, maybe codes or some other thing, that might need a real-time machine learning model and that's what streaming comes into place, which can stream data to a model in real-time, we can use this model to make decisions and stream it back to our product. It's largely between batch process and real-time processing. Different systems help you do both of those things.

[0:50:39.2] JM: Okay, interesting. Let's zoom out a little bit. Crunchbase tells me your company is 50 to 100 people. Tell me about the experience of scaling Ethos.

[0:50:50.7] LW: That's accurate. We're at about 80 people right now. Two years ago, I think we were maybe less than 10 people. It's been very exciting, the ability to scale the company. That also comes with the number of challenges. To be totally candid, Peter and I are managing such a large team for the first time and we're very fortunate to have brought on exceptional experienced leaders like Vipul, who has managed a large team in the past.

Growing the team is super fun, because now we're able to tackle bigger and larger problems than when it was just eight of us. That's been super exciting and something that we're looking forward to growing in the future.

[0:51:37.2] JM: How's the company structured at this point, in the sense of what are the different teams? Because you've got these interesting data science problems. You said you have a role, what was it? Product engineer? No –

[0:51:50.2] LW: Product scientist.

[0:51:50.8] JM: Product scientist. What is a product scientist?

[0:51:53.5] VS: Product scientists are folks that have data experience, but also deep product experience. Particular data scientist or data engineers would largely look at the problem from engineering point of view .They would be very excited in the scale of the data, the processing of the data, building models in a way that that can derive some output, or solve a problem, but product scientists are skilled in understanding that data and relating it to how to make changes to the product. That's a very product function.

Our head of product, Gokul have deep experience coming from Facebook, Instagram, Snapchat. One of the things that these companies have done really, really well is use product scientists to use the data that the engineering is building and make the product constantly better by unlearning from it. That's what product scientists are. What they worry about on a day-to-day basis is how is our product funnel, what stages are there in the funnel, what attributes and thinks we should be looking at. Is an experiment doing better on desktop versus mobile? Is a particular experiment doing better in certain segments of users versus not doing certain segments of users?

These deep insights are extremely useful for us to continuously create an experience that our users will love. It's not just quantitative, because there was a big qualitative piece to it, which is where a design theme plays a big role in constantly talking to our customers, running user testing and also keeping – understanding how our users are qualitatively talking about our product. You take this quantitative and qualitative data that we gather from various sources and make data informed decision. That's what product scientists are all about.

[0:53:38.6] LW: Then to go back to your earlier question of how our company is structured in general, we have different functional areas, so we have engineering, product, design, customer experience, marketing, legal, so on, so forth. There are also cross-functional pods that work together.

For instance, our growth pod includes not only engineering and product and design, but also includes our customer service reps, so on, so forth, and our analytics teams. That's how we think about the company in slices of the customer experience. That's a major factor in how we structure our teams.

[0:54:22.7] JM: Hey, Vipul. What's the hardest engineering problem you've had to solve in working at Ethos?

[0:54:29.5] VS: When I started, the engineering team was six people. Now we are getting closer to 30. To me, when I think about problems, I think about in three ways, which is product, people and process. My role is to think about all of those three things at the same time, which is as we are scaling our engineering team and hopefully from 30 to 60 very soon, what processes are working and what processes are not working? A certain set of agile frameworks, which might work at when you are 10 will not work when you are at 50. How is our culture evolving along with that? That's the process part of it.

The people part of it, what kind of people do we need at different stage of the company? Earlier part, you are extremely motivated to ship product as fast as you can. Little bit later part, you want to do that, but also ensure that the quality is high and you can ship fast, which brings a lot of architectural problems into the place.

Currently, we are building an architecture team which will take our application and provide different services on top of it, so that different teams can take ownership of their part and move really, really fast. The current problem we are taking is our app is currently in breaking down into different services is a very, very interesting problem to solve. Then comes the product part of it, which is thinking about now this application process.

It looks very simple when users interact with it. If you look at behind the scene, it's a very broad and deep tree. Because based on the answers you are giving to the questions, your next set of questions will appear. This could be an extremely slow process to traverse that tree. We want to make sure that that is as fast as possible. We're constantly thinking about how do we give this experience to the users in really, really fast way? That's a very interesting problem the team is working on right now.

[0:56:22.9] JM: All right, last question. Lingke, tell me something non-obvious about the life insurance business that you would not know if you had not created two businesses in this space?

[0:56:39.1] LW: I think that's something I wouldn't have known without having dug deep into this industry is actually how much technology can change what we do in this industry. I think it's easy to say from an outside perspective, “Hey, if we build brand-spanking new systems that it will make things more efficient.” I think that's a very surface-level way of looking at it.

At the end of the day, what I've come to realize is that yes, building a end-to-end system from the ground up is super important. Two, it's really the data and it’s really the understanding of our data and our models that will allow us to deliver a phenomenal customer experience at the end of the day. Really what that takes is redesigning the end-to-end stack of life insurance from a tax standpoint, from a process standpoint, from an experience standpoint, from the ground up in order to succeed.

[0:57:39.9] JM: Okay, guys. Well, thank you for coming on Software Engineering Daily. It's been really fun talking to you.

[0:57:43.8] LW: Thanks so much. Appreciate the opportunity.

[0:57:45.6] VS: Thanks, Jeff. Thanks for having us onboard.

[END OF INTERVIEW]

[0:57:56.8] JM: Software Engineering Daily reaches 30,000 engineers every weekday and 250,000 engineers every month. If you'd like to sponsor Software Engineering Daily, send us an e-mail, sponsor@softwareengineeringdaily.com.

Reaching developers and technical audiences is not easy. We've spent the last four years developing a trusted relationship with our audience. We don't accept every advertiser, because we work closely with our advertisers and we make sure that the product is something that can be useful to our listeners. Developers are always looking to save time and money and developers are happy to purchase products that fulfill this goal.

You can send us an e-mail at sponsor@softwareengineeringdaily.com. Even if you're just curious about sponsorships, you can feel free to send us an e-mail. We have a variety of sponsorship packages and options. Thanks for listening.

[END]