EPISODE 580

[INTRODUCTION]

[0:00:00.9] JM: When I buy a mattress online, I pay for it with my credit card, behind the scenes, a complex series of transactions occurs between a payment gateway, the credit card company and a few banks. There are problems with this process. It's slow, it's complex, it involves the synchronization of several different parties. Also, I can just buy my mattress.

Some consumers will not want to purchase that mattress because they do not have the cash upfront and the lending rates that they get offered are higher than they're willing to spend. If these consumers were presented with more intelligent loan rates, the lender could still make money, the mattress could still make money and the consumer would get a new mattress. It's a missed opportunity all around.

Affirm is a consumer financial services company, their first product offers loans to consumers, making purchases. In today's episode, Affirm CTO Libor Michalek explain how Affirm decided to build this product and what they've done to scale it. The conversation took me by surprise because Affirm was started by Max Levchin, who is a cofounder of PayPal and I assumed that when Affirm was created, they already knew exactly what they were going to build because Affirm is a payments company and Max has knowledge of the payments industry going back several decades.

In reality, Affirm started out with more vague ideas around what they were building. They spent some time running small experiments and they just looked for product market fit just like a bootstrap startup would have. It was inspiring to know that even as an experienced team, they were willing to go through the humble process of searching for a product within a space that they were deeply familiar with.

We didn't get to all the questions I was planning to explore but I hope to do another show about Affirm in the future. We've done shows in the past about engineering at other payments companies like Stripe and TransferWise, as well as other financial technology like block chains

and automated trading and if you're looking for all 700 of our episodes of software engineering daily, you can check out our apps on the iOS or android app store.

You can listen to those episodes at softwaredaily.com. We've also got tons of episodes on business and distributed systems and lots of other topics. If you want to become a paid subscriber to software engineering daily, you can hear our episodes without ads. You can subscribe at softwaredaily.com and all of the code for our apps is open source. If you're looking for an open source community to be a part of, come check out github.com/softwareengineering daily.

With that, let's get to this episode.

[SPONSOR MESSAGE]

[0:02:53.7] JM: Apps today are built on a wide range of back ends from traditional databases like Postgres to MongoDB and elastic search. To file systems like S3. When it comes to analytics, the diversity and scale of these formats makes delivering data science and BI workloads very challenging.

Building data pipelines seems like a never ending job. As each new analytical tool requires designing from scratch. There's a new open sourced project called Dremio that is designed to simplify analytics on all these sources. It's also designed to handle some of the hard work like scaling performance of analytical jobs, Dremio is the team behind Apache arrow.

A new standard for in memory column or data analytics. Arrow has been adopted across dozens of projects like pandas. To improve the performance of analytical workloads on CPUs and GPUs. It's free and open source, it's designed for everyone from your laptop to clusters over 1,000 nodes. Check out Dremio today at dremio.com/sedaily.

Dremio solved hard engineering problems to build their platform and you can hear about how it works under the hood by checking out our interviews with Dremio CTO Jacques Nadeau as well as the CEO Tomer Shiran. At dremio.com/sedaily, you can find all the necessary resources to get started with Dremio for free.

I'm really excited about Dremio, the shows we did about it were really technical and really interesting. If you liked those episodes your you like Dremio itself, be sure to tweet @dremiohq and let them know you heard about it from software engineering daily. Thanks again to Dremio and check it out at dremio.com/sedaily to learn more.

[INTERVIEW]

[0:04:54.7] JM: Libor Michalek, you are the CTO at Affirm, welcome to Software Engineering Daily.

[0:04:58.4] LM: Thank you, I'm happy to be here.

[0:05:00.0] JM: I want to start with a basic user problem and then we're going to work our way towards engineering so let's say, a consumer wants to buy a mattress online or any larger purchase that takes a lot of money. The consumer does not have cash to pay for that mattress immediately and if we're putting ourselves in the world of traditional consumer finance, the main option to pay for that mattress, if you don't have cash is with a credit card.

You've got Visa and MasterCard and MX and Discover. What are the downsides of using a credit card in that situation?

[0:05:39.6] LM: The main downsides is that the ultimate all in costs at that point are not presented in a clear manner and to a certain extent, are difficult to know based on how else you're using that credit card. Once you start revolving on that credit card, all your other purchases that you're going to put on it are going to carry interest as well and your pay down plan is not clear and the costs will be opaque to you.

The logical thing to do, if you were to put it on your credit card is at that point to stop using that credit card entirely. Now, that is not a straight forward option for most people. When they go in to making a purchase and they know that it's going to be something that they have to pay for over more than one-month period based on their cash flow.

The logical thing would be to use something where they do know exactly how much that one purchase is going to cost them over some period of time that fits their cash flow. Similar to how a lot of us wouldn't put a car on a credit card, look for a number of people, any large purchase that goes beyond what you can afford in one month would fit the same criteria as I want to know upfront how much this is going to cost me all in. That's where we come in.

[0:06:53.8] JM: In that world where customers are interacting only with the traditional credit card systems, what are the other big inefficiencies?

[0:07:03.6] LM: The other big inefficiencies that we like to think about are the pricing on the per purchase bases is one of the big ones. By that, I mean, the merchant has various ways to – merchant specific underwriting allows us to price that transaction specifically to not just, where's this user coming to us from the merchant but what is the merchant willing to help subsidize in those purchases?

For example, a large margin products like mattresses, we can work with the merchant to ensure that user gets the best deal possible for example things like 0% interest on those purchases for those products. Similarly, users that wouldn't be able to get credit or otherwise, we can work with the merchant insurer that they can get credit at reasonable prices and similarly improved interest rates or even things like you know, partial 0% or cash back rewards, various ways.

Whereby looking at that exact transaction, we can make sure that users are getting the best deal. This is different than where the user normally would – in traditional credit cards, you would move from your bronze card, your silver card, your gold card or your platinum card, one decade at a time as you finances and credit improved over time, this is something that we can do on the spot for every single transaction and make sure that that point in time, based on the most up to date credit information and merchant information that we have.

We can price that deal optimally for both user and for the merchant.

[0:08:46.8] JM: In the example with the mattress, you eluded to the fact that a mattress is a high margin item, each individual mattress purchase is going to lead to a lot of profit on the part of the mattress creator. If there's a situation where a customer cannot purchase that mattress

because they don't' have the cash handy and the firm can facilitate the transaction by providing an alternative financial purchasing process then you can potentially take out margin from the merchant.

While keeping the deal that the customer has the same. Opaque, that process is opaque so the customer just gets the perception that it pays zero interest, that's remarkable, whereas the merchant is saying, well that customer would not have been able to purchase the mattress otherwise, that's remarkable, even if we make 75% margin instead of an 80% margin then we're still –

You know, we're still gaining capital that we would not otherwise gain from that transaction and then Affirm gets to pick up the difference, is that accurate?

[0:10:02.8] LM: I wouldn't say high margin, I would more specifically say that the – I mean, the mattresses is actually a very competitive and so it's hard to say that it's high margin. It obviously, the merchant has some amount of budget for things like marketing, for their own take home as well and they want to be able to ensure that the user is getting the best deal possible for things.

In a mechanism that actually matters to the user that users respond to and so, the most obvious is these are larger purchases and the typical customer isn't able to pay for it in a lump sum manner. That small difference of you know, taking it from 10%, 15%, 20% interest for a year, you know, something that would take somebody a year to pay back or six months.

Being able to move that to 0% is something that users see as a clear benefit to their pocket book and sort of bang for buck from the merchant's perspective. It's much more effective than you know, plastering a bard or you know, the sub way with ads. Giving that directly to the user who is already interested in that mattress makes much more sense for them and that's where we like to facilitate that transaction.

By making sure that our users are getting the best deal possible.

[0:11:27.3] JM: This is of course not mattress sales daily or financial engineering daily, it's Software Engineering Daily and we will get into the software aspect of things but I wanted to start with this micro cosmic example just because looking into some companies like Affirm and like stripe and these other newer fin tech companies.

Wealthfront, you see how many opportunities there are for fin tech players with some sophistication in technology, to come in and improve the world of consumer finance. We've given this micro cosmic example of the mattress purchasing processes but there's a much wider array of problems that the average consumer encounters when interacting with the financial system.

What are some of those other big problems? Give me a larger idea of the scope of what challenges Affirm is trying to tackle?

[0:12:23.4] LM: Well, I think probably the sort of the highest level, we think about what is the opportunity for a company like Affirm within consumer finance. We look at where does a lot of the profit and consumer finance come from. A lot of it comes from the unwanted or the undesirable or the unexpected happening to the customer.

This level of asymmetry between users that are attempting to manage their finances, accomplish their financial goals and where a lot of the money that's made from those customers is coming from seems that asymmetry seems right for disruption and for better product, better consumer experience to come in.

That can be in the form of credit and the form of payment and the form of savings, investing, all of the above. We decided to start with credit because that, as a clear compelling use case, even when as a company, we were small and just getting started.

The clarity there for us was, okay, you know, if you were thinking about this from sort of first principles, what would that look like and it's obvious to us that you would put forward the upfront cost that the user was going to pay. You know, we all know that if you're going to pay for something over a longer period of time, it's going to cost you more.

Having that level of clarity, knowing that there's no late fees, that there's no going over this amount and in fact, if you pay off sooner, you come in under that but that there's a cap and the user can say, is this a thousand dollar sofa worth an extra $70 to me to have it a year earlier, that seemed like a clear value proposition to the user, the user's obviously started using that product and the merchants like it for similar reasons.

There is plenty of asymmetry and information that exists and I think that's where a lot of the opportunities are. Where we can – has on credit, has on savings, as on credit debt pay down, you know, getting gout of debt, all of these elements are – there's a lot of information that about the user out there that the user has about their credit, about their finances and using technologies, using software to help them better understand exactly where they are in that ecosystem.

What is 12 months out look for them, how do they improve their financial situation? It's an interesting dynamic where I think there's a lot of room to have information parody with your customers where you know, the company's understanding of that customer is improved by that customer understanding how that information is being used and how that information actually informs those decisions.

Some of the ways we think about the problem going forward is, how can we improve that sort of by directional information feedback loop between us and the customer to further refine them, further improve both what the customers use and what we understand about their ability to manage their finances, their ability to take on credit or you know, savings, helping them achieve those goals.

[0:15:40.3] JM: The first product that you focused on was this credit application. Explain what the first version of the affirm product did and what the product surface area was?

[0:15:53.3] LM: There is many iterations of the first product before we found something that was both compelling to the customer and the merchant. Obviously you need to be able to distribute your product for it to gain any traction and the first end iteration of the products were entirely about solving for that.

The first iteration was a stripped down deferred interest, not differed interest, deferred payment product. It was almost a simplified invoicing product and less of a credit product where the user would go through a purchase flow and there was a simplified purchase flow compared to entering your credit card information.

Especially on a mobile device and where we would just get their phone number and a little bit of information. Then when they were back at a computer a couple of weeks later, a month later, be able to send them an invoice for payment.

Initially, the idea was okay, how do we simplify that process? Then as we went through version after version, to see what was actually resonating with people, it became clearer and clearer that people were beginning to use it as a credit product and wanting to pay back in multiple instalments.

That really quickly then, sort of said, I see, the thing that they're responding to is not just the convenience of how to make purchases like this on a mobile device but actually pointing towards that, the users themselves pointing us towards, well credit for us is broken, you guys seem to be going down this path, help us solve this problem.

[0:17:32.7] JM: That's a fascinating way of exploring the domain of consumer finance, you have some set of merchants that you were working with that as part of the checkout process, the user was presented with the option of paying with Affirm and if you pay with a firm in the product market fit search period of time, from the affirmed point of view.

You just enter your phone number and you get your matches paid for or whatever the product was and then it's like the mattress comes and you're like, that was magical and then Affirm behind the scenes is like, okay, what are we – how are we going to turn that into a product?

[0:18:12.7] LM: Exactly. That was a lot of it is like exploring the space of convenience and you know, what doesn't work in that space today and that sort of search started really pointed the avenue towards both of the – from the customer's point of view, the broken credit but also from the merchant's point of view where that convenience and that new type of credit that customers were responding to was a value ad for both of them.

That was really where we're like okay, this makes sense. From that point, where we started to tune the product. Okay, what do we need from the user without making it a part of some checkout process, what do we need to actually provide credit over longer periods of time and obviously from the merchant perspective, how can we help quantify for them exactly.

Who are these users, how are these users responding to this product and therefore, you know, getting them excited about it as well.

[SPONSOR MESSAGE]

[0:19:16.8] JM: Software work flows are different at every company. Product development, design and engineering teams each see things differently. These different teams need to collaborate with each other. But they also need to be able to be creative and productive on their own terms.

Airtable allows software teams to design their own unique work flows. Airtable enables the creativity and engineering at companies like Tesla, Slack, Airbnb and Medium. Airtable is hiring creative engineers who believe in the importance of open ended platforms that empower human creativity.

The mission of air table is to give everyone the power to create their own software workflows. From magazine editors, building out their own content planning systems, to product managers, building a feature road maps, to managers managing livestock and inventory.

Teams at companies like Condé Nast, Airbnb and we work can build their own custom database applications with the ease of using a spreadsheet. If you haven't used air table before, try it out, if you have used it, you'll understand why it is so popular. I'm sure you have a workflow that would be easier to manage if it were on air table. It's easy to get started with air table but as you get more experience with it, you will see how flexible and powerful it is.

Checkout jobs at air table by going to airtable.com/sedaily. Airtable is a uniquely challenging product to build and they're looking for creative front end and back end engineers to design systems on first principles. Like a real time sync later, collaborative undo model.

Formulas engine, visual revision history and more. On the outside, you'll build user interfaces that are elegant and highly customizable that encourage exploration and then earn the trust of users through intuitive, thoughtful interactions. Learn more about air table opportunities at airtable.com/sedaily. Thanks to Airtable for being a new sponsor of software engineering daily and for building an innovative new product that enables all kinds of industries to be more creative.

[INTERVIEW CONTINUED]

[0:21:36.0] JM: I just want to comment on the humility of that approach where you've got, you know, yourself and Max Levchin, the CEO between you two, have experienced it. PayPal, at Google, at Slide, the company you started before that and instead of taking an opinionated approach to her is our perspective on the market and we're going to build this product, even with the volume of experience that you had, you took the humble approach of let's really try to approach this space with wide eyes.

I find that that's rare, it's kind of inspirational to think about that.

[0:22:19.8] LM: I think for both of us and for the team as well that we've put together, that's been sort of a driving ethos over the course of it was you know, you have you prejudices of how things work in the world.

You know, I think you do it enough times, you realize that that's only a starting point, you actually have to understand, you have to discover the process of why do users respond. By users, I mean, the customers, the merchants, providers of capital, how do they think about these problems.

Like I said, you have starting points for ideas, you know, the unfairness, the asymmetry, all of these things of the existing consumer finance. But you also have to ultimately put those to a test

and think about what does everybody else actually think about this and so, you know, one of the things we talk a lot about here as we build out the team is the idea of building a startup as a journey into the unknown.

That itself requires a level of humility. We look for people that excites them, you know, we're going to go, we don't know the answers but we have the scales and we have confidence and our capabilities to go figure this out and make it happen and see where the journey takes us and people who are excited about that are the people we love to work with.

[0:23:42.3] JM: By the way, one other question on this, product market fits search space, who are the merchants that you were working with from the early days, can you comment on that? Our first merchant. The first merchant that really took a chance on us was 1-800-flowers, they were the one that a lot of exploration with and then went from there to know, work with Casper on the mattress side, was one of the first ones we worked with real and trades early on.

These were the, they're pretty distinct merchants. That helped us also think about what does the product look like through different lenses. That journey continues today, we start to work with companies like Expedia and travel or you know, some of the large, our offline retailers.

They're helping us to continue to think about, what does this look like in different merchant landscapes for different use cases. What type of users that we haven't seen before, you know, show up at those merchants and what does that product look like for them? That really helps us continue to prioritize the world map build out, the world map and explore it.

[0:24:56.4] JM: Also, I don't know if this factored into your merchant selection but it seems like flower purchasing or mattress purchasing are domains where there is a little less fraud on the purchaser side.

[0:25:08.3] LM: That's definitely true. I mean, we did and continue to find that in our fraud analysis as well as in our underwriting analysis that the merchant selection matters a lot. We sort of had that theory that that would be true, that people respond differently if they're buying durable goods versus different types of consumable goods, different types of durable goods.

That did play out that way and it probably even more so than we really expected and so on that way, pretty fortuitous with the initial selection.

You know, when you're that small, you're going to – it's more about the merchants taking a chance on you than being picky about – overly picky about, we're going to help out.

[0:25:48.8] JM: Again, it sounds like the early product surface area was basically a little signup form where you like entering your phone number and then behind the scenes, you're just like paying probably using some legacy payment integration software. When did you actually like have to do it, engineering work?

[0:26:12.3] LM: Well, I mean, that's actually, I mean, that is one of the things that's exciting, this is the fifth company, fifth startup I've worked on and you know, that's the exciting part for me as an engineer as you go, you pivot form the product market fit, you know, your code base and your team are really optimized around the idea of, let's use this small team and let's explore really quickly, lots of ideas and you're really kind of building yourself to around that idea.

Then, when you stumble upon it or and you're like, this is it, let's figure out how to scale this, obviously, you know, scaling the team scaling the business but it also means scaling the code base because you're really expending on multiple axis, everything from new features, more robust features to handle all the edge cases.

Software that can support actually all of these engineers showing up and working on the code base, scale obviously from just the perspective of more users. That ultimately going from this prototype to this large sustainable business is an engineering challenge and it's basically engineering fast forward. I always tell, especially younger engineers but really, anybody who is in that.

If you want to sort of have the quintessential software engineering experience of how do you go from something optimized for quick pivots to robust large scale software and you want to have that experience as fast as possible, you know, go to a company like us or anybody else really who is going through that transition and the faster they're going through that transition, the more of it you'll sort of get from the fire hose on.

That's been fun for me and it's been consistently fun every time I've done it.

[0:27:56.1] JM: Obviously, there's a number of different dimensions to the product at this point. You've got a ton of data that's coming through the system, but you've also go the product itself, the user facing product that people are integrating, people interfacing with.

You've got the back in infrastructure. The product that people are interfacing with, you've got the back end cloud infrastructure where the transaction, certain transactions are occurring and then all this data is accumulating over time and you've got to run machine learning jobs on that to build a good risk model, to be able to decide who is worthy of certain credit products.

I think before you get to the scale out of the cloud infrastructure or the detailed risk model in machine learning discussions, you did have to build out this user facing product, you had to build out the basic skeleton of some back end infrastructure. Can you take me through that early product, once you found the bit of product market fit that you wanted, but you had to build out the basic Ruby on rails app or Jango app or whatever was –

[0:29:08.6] LM: Flask app. Yeah, you know, you got Flask, very kind of model view controller sort of paradigm sitting on top of OR map or using SQL coming for that, really basically running the side of the online interactions between us and the merchant and the customer.

You've got a few batch pipelines that are sort of doing, you know, your nightly messages and your billing and your auto pay. Using Selery for some differed tasks, that's kind of full to the nexus of kind of where we started and including the online divisioning models and yeah, that's where it started and that's really kind of what we've been iterating on.

You know, one of the takeaways from past experience has been. well you are growing this business quickly. You are not going to stop to rewrite all of your code partially because you are moving quickly but partially because one of the things that is really, really hard in a fast growing environment does contacts on why are things the way they are, why these decisions are getting made and a lot of times the only thing that really embodies that sort of tribal knowledge of why are things they are is the code itself.

And so when you think about rewriting large code bases, you're pretty cow fit and you should be relatively confident that you will get the primary use case as correct but all of the edge cases where I did that that have been tweaked and optimized during this growth, those are the things that you're likely going to miss. I mean if you try to rewrite wholesale and those are the ones that you worry a lot about because especially in our case.

Where the first feedback that we have on a mistake with a model, a credit model is going to be 60 days out when we really start to say, "Hey this decision prediction versus what's happened is off." Those are the ones that you get to worry a lot about and so you spend a lot of time at least we did initially saying like, "Okay the first thing we're really going to focus on before as we are growing is let's get monitoring and alerting it placed," both operational monitoring is the thing working the way we expect it online.

And the batch by points but also business monitoring. So like are the results coming in the way we think of they are and alerting on top of that and then the next big steps are okay, now think about how do we pull this code base apart into constituent pieces and then how do we think about iterating on each of those as we go and really sort of you know these are overlapping steps and approaches but sort of taking it from there versus okay well we have this codebase.

It got us to here, it is obviously going to have a tough time supporting a hundred, 200, 300 engineers. Let's throw it out and start over again. It's more of a how do we kind of walk this forward while still making progress on the product.

[0:32:11.0] JM: Of course, so that early product it sounds – were you able to dodge integrating with traditional banking infrastructure?

[0:32:21.5] LM: No, we definitely said that at the Nexus of a lot of partnerships. Yeah, we have a lot of data providers those are probably the most lobbyist ones everything from the traditional credit bureaus especially on the front side, that non-traditional sources. I mean non-traditional sales but probably more fancy that it is but things like phone porting records. When was the last time this number was put and things like that and then on –

[0:32:44.4] JM: Sorry, phone porting?

[0:32:46.0] LM: Yeah, you know when the last time this phone number changed hands, how long are those and things.

[0:32:49.5] JM: Oh okay.

[0:32:50.5] LM: And on the bank side yes. Definitely a lot of traditional partnerships. Everything from how we send money to merchants, how we get money from capital providers and from our own bank accounts, how do we get money back from the users and how does this all flow through the different places and it needs to go. A lot of that is very traditional funding operations and interacting with them and yeah that's been definitely a learning experience as well.

[0:33:21.4] JM: So what's the release process for a piece of consumer financial software? Do you have to go through audits, is there certain standardized testing processes you have to go through?

[0:33:37.7] LM: I mean yes but nothing surprise – nothing that would look surprising to somebody who's been writing software in Silicon Valley over the last 10 or 15 years. You know one of the things that we thought and spend and continue to spend time out upon is when we think about what are the compliance regimes and what are the regulations about data access and testing and disclosure compliance, all of these things.

We think about, well how does that actually map out the good software engineering practices? Everything from code review to automated testing to unit testing, functional testing, end to end testing and it's really been a process of, "All right how?" Obviously this compliance regime or this rule or what, it is trying to achieve something practical. What is it trying to achieve and how would you actually translate that into something that logically makes sense to an engineer?

And then working with our compliance team to make sure that the two actually map together and that we are reading it correctly but that's really been how we thought about it and so really, really early things like code review. So that every piece of software gets looked out by multiple

people before it goes into production, that the released is signed off and what is actually going to production is signed off by somebody different than the person who wrote the software.

That we have good visibility into could anyone have touched it before that happened or do we have confidence that this really was looked up by multiple people before it went out, who has access in the production environment and again, when we are moving fast there's time where you need to access to write production data or production code, production environment but being able to really facilitate that with, "Okay let's make sure we actually have visibility and what that person is doing."

And somebody else can look at it what that person did out of band and it has been a process of translation. So we really rarely got to the idea of code review, automated release process, release signing, unit testing, functional testing so including videoing some of our testings. So that legal could look, things like that where at previous companies, we wouldn't get to some of these practices until we're 50, 60, 100 engineers. These are things we were doing when there was just 10 of us.

[0:36:06.9] JM: So what are the important ways that the product has advanced since the early days of Affirm?

[0:36:13.4] LM: The product, a lot of what has happened to the product is the – I mean sort of underneath the covers a lot has happened with how we actually worked with our capital providers in terms of how they understand the product and how they help fund the loans that we produce, the credit we push out, how we think of the models and the data science underwriting and fraud siphons that we are building out.

Really how we are thinking about first party and third party data for those models and separating that out from right the rest of the codebase that is moving fast and on the product side, it's been how does the merchant and the customer better understand this product and how do we work in more environments and so creating more visibility and more products for the merchant to work with so that they can tune the product and understand it how it relates to their business better.

And then on the customer side, it's been much more around clarity customer support and what that looks like as well as once the user has used the product for the first time, how they can use it in multiple situations. For example, we have built out an app where we are generating one time use credit card number for the user to use anywhere that they want to. So that they can take this experience that they had at a merchant and basically go down the other merchant and have a similar experience.

And to facilitate that, obviously the rest of the world runs on credit cards. What we do is the user will tell us what is it that they want to buy, where do they want to buy it, then how much it cost and we'll generate a onetime use credit card for them, a virtual credit card to go make that purchase. Similarly, helping those users better understand their overall finances and so having them link their other credit card accounts within the app.

To then paint a picture of how much are they paying on fees on their credit card, how much are they paying in interest and really help them understand where Affirm fits into that picture including when they should be buying on their debit card versus credit card is when the firm makes sense. When it actually doesn't make sense to buy something at all and probably the best bet for them is to just pay down their credit cards and focus on them.

[SPONSOR BREAK]

[0:38:48.9] JM: Failure is unpredictable. You don't know when your system will break but you know it will happen. Gremlin prepares for these outages. Gremlin provides resilience as a service using chaos engineering techniques pioneered at Netflix and Amazon. Prepare your team for disaster by proactively testing at failure scenarios. Max out CPU, black hole or slow down network traffic to a dependency, terminate processes and hosts.

Each of these shows how your system reacts allowing you to harden things before a production incident. Check out Gremlin and get a free demo by going to gremlin.com/sedaily. That's gremlin.com/sedaily to get your free demo of how Gremlin can help you prepare with resilience as a service.

[INTERVIEW CONTINUED]

[0:39:47.8] JM: The financial identity side of things seems important if you are starting to move beyond having specific merchants because if you only have specific merchants that you are providing credit to consumers on within the domains of those purchases, your operations are probably a little bit simpler but if you start to get into the game of providing financial services around any kind of purchase, then it becomes more – well I guess it becomes only a question of who is this customer and are they credit worthy?

[0:40:24.3] LM: Right, yes and it definitely gets more and more complicated. There is obviously some information about where and when they use that instrument when they are shopping but that is one of the things that is actually beneficial to having this incremental relationship that builds overtime, right? Each interaction with them provides more information about how they think about their finances and how they urge their finances and to what extent is it compatible with what we're trying to do.

And what we are trying to provide and build on that overtime but yeah, what is interesting about that it's engineering perspective is it starts to shift because of the weight of the decisions, each decision that we make from third party data to first party data and I think normally people think about, "Well you know first party data" meaning our data that sounds easier because it's your data and in some ways, it's actually from an engineering perspective.

It is actually especially in a company that is moving quickly, it's a little bit harder than third party data. The third party data is at least in its interfaces and how you interact with it is fairly static, right? These companies have been around for a long time. They're in the business of providing APIs into this data and so it is a fairly slow moving affair. Our own underlying software and systems and data is moving quickly and evolving quickly as we build products.

And as we iterate on them and you know thinking about what are those internal interfaces look like. Like I said as the engineering team doubles every 12 months, how do those interfaces look and how do we make sure that where we have assumptions about how a piece of data is going to work that not only do those assumptions continue to hold over years as these models sort of pick up steam but that that is codified in the interfaces themselves and that we teams give dependencies to other teams.

Versus teams taking dependencies on pieces of it, things like that and it becomes fascinating for at least I think fascinating engineering challenge of how do you build something that grows quickly.

[0:42:33.9] JM: Yeah, okay so the third party data question versus first party data question, I hadn't thought about that. So the first time a customer makes a purchase through Affirm if they're buying from one of these limited scope of merchants like a mattress company for example and this is their first interaction with Affirm and they give their email address, the last four digits of their social and their phone number and their name?

[0:43:05.2] LM: And their name and their date of birth.

[0:43:06.3] JM: Name and date of birth, so with that information you could go at a third party data providers that could say, "Here's credit report, here is the most recent time the phone number changed hands, here is the average credit profile of a millennial," for example, things like those. Those are well established in terms of some of the signal that you can get from those different things. You can also enrich it with all kinds of other data providers where you can get a really, really rich signal around somebody just from those essential four pieces of information.

[0:43:40.3] LM: Yep, five pieces yeah.

[0:43:42.4] JM: Five pieces of information right and then overtime, you develop a domain specific understanding of how rapidly they pay for something. So then you could give them a more tailored set of credit opportunities based off of your first hand interactions with them and how they match up with perhaps some collaboratively filtered other set of customers that look like this type of person that has also bought mattresses.

[0:44:13.2] LM: Correct. Yeah, we start to build out how does that behavior – what does the behavior look like and what does it do about predictability relative to similar looking people and how they've behaved overtime as well.

[0:44:26.5] JM: Okay so tell me what is hard about building the machine learning models around this process? One thing you hinted at was the question of the time horizons for these. You know you build some model around a limited time horizon because the firm hasn't been around for that long. I mean it's been around since 2012. That's long in the time span of startups. It's not long in the time span of a person's financial lifespan.

[0:44:54.0] LM: Exactly and in that sense, it's been also literary a learning experience, machine learning experience and that iterating on the models. So as we build up our own first party data, it's retraining the models on more and more data. One of the things that is a challenge there is where is the codebase. We look back two, three years of user's information that we have of how they interact with our product. Well obviously our code bases changed tremendously in that time.

In the time that the team has gone from 10 engineers to a 100 engineers, you can imagine the codebase going through quite a bit of change and we plan on similar, this sort of degree of change for the next careers and so it is a challenge to think about how is the model interpreting this data from three years ago versus today and how is it going to look three years from now and the idea of sort of having this temporally agnostic data platform.

Where you can go back in time, run a new model against what the data looked like at that point in time, see what the decision is like and similarly, go back in time with a new piece of data and think about like, "Well what would have happened if we had to use it at that point in time?" and then enough so that we'd be able to train and validate those models and then put it.

[0:46:13.3] JM: Well so back testing.

[0:46:14.2] LM: Yeah, a lot of back testing and validation and the other part of it is also looking at you are obviously you don't want to just retrain unjust data that the previous model told you was a good customer. You obviously want to look at your false negatives as well as false positives and be able to train against that data as well and part of this is as we think about building these models, having them look at more data, having them produce sort of ritual results as well across the space.

And that means putting out models that has a fairly frequent cadence and being able to accelerate not just keep the same pace as the data continues to grow but actually accelerate that on the range in which we can do this.

[0:47:02.1] JM: The penalty for getting a model wrong, is that going to be that you have a higher default rate on credit?

[0:47:14.3] LM: Exactly.

[0:47:14.9] JM: Okay.

[0:47:15.9] LM: I mean think of them in a few forms. It can come through on fraud. It can let through people that we shouldn't have. The fraud one is obviously frequently as I think top of mind but one of the – I mean the good and the bad thing about fraud is that it moves pretty quickly. So it moves quickly meaning you can lose a lot very quickly but you can also detect it very quickly and jump in and intervene. On the credit side it's the opposite.

So a lot of things if you get the level wrong, probably the worst is when you get a suddenly wrong where you are off by a few percent and it's not obvious until a few months out as that you have been apt. You have been lending at a slight lose which at our scale can be millions of dollars and because you have been doing it for potentially months that even if putting out a model now, that still has to work its way through the system.

So those more subtle deltas are the ones that we spend a lot of time thinking about and keeping an eye on and given the model where you have 40, 50 plus signals that go into it all with various weights, being off in one or two of them can shift things pretty subtly.

[0:48:35.8] JM: There's concern around bias in machine learning datasets and how that can affect opportunities for people who might fall into certain biased categories. Do you do anything to control for bias in the datasets?

[0:48:54.6] LM: Well we stay away from signals that would kind of lend themselves to obvious bias. You know we won't put somebody's name or somebody's zip code into the model, things

that are traditionally associated with especially socioeconomic bias in these biases and we do also review the results and including independent review where we look at, “Okay well what did we do and how does that – how did that sort of come down on different groups.”

And then we sort of benchmark that against how does the rest of the industry performed, see that it’s exact, make sure that we are going in the right direction on some of these models. So we review, we look at it both internally as well as sort of through external providers to see how we are doing. We feel like we have been actually doing quite well on this in the sense that looking at a lot of signals about how are people actually paying and repaying.

How do they behave based on where they’ve come from and what are they doing, really does provide an avenue where we are not dis – people who have not had access to traditional credit and that becomes sort of a pernicious, persistent problem for them. We then break out of that cycle and so very quickly being able to establish a credit history with us and because it is on a per transaction basis, being able to really quickly from their perspective as well as ours be able to iterate them.

And not going through this, “Well you have this credit card,” where you have to put a down payment on the credit card itself and you are going to be in that situation for two years and then you get a slight bump in your limit and that is another five years. Being able to enter not just from our model data perspective but even from the user experience perspective right? Of just getting this feedback and being able to ingest it, ensure that we are as up to date on our decisions as possible.

I think helps ensure that there is a better feedback mechanism for from the customer as well that you know, positive financially responsible behavior leads to better results in the product itself.

[0:51:17.9] JM: Yeah and I mean not to mention you will get more opportunities for customers that way because often times the legacy providers have not fact – I mean legacy providers have their own models and they are probably more likely to factor in bias because they are more naïve models and you have other companies that had other companies on the show who are doing this kind of thing, machine learning for short term high interest loans for example.

Or maybe even short term low interest loans and they find that conventional financial service providers are often not serving certain people who biased data sets may flag as not deserving of credit for invalid reasons and so that you can build a very good business off of these types of different, bias differentials.

[0:52:14.1] LM: And this does come in too where we set aside budget too within our learning program to actually look at users that are even the model says, "You shouldn't, you know this user can't be approved profitably," letting some of that and looking at how those users actually perform to ensure that we are expanding the scope of our underwriting and our fraud detection to really look at a broader population and see how they perform versus right to the model just telling you.

Basically having confirmation biased where the model calls through, only these people are going to perform well and if you only approve those people, then the next model you train is going to tell you basically the exact same thing, right? And so making sure that that we are actively on seeking out a broader population and including underserved populations to see, "Okay what else are we missing out there?" Obviously there is a whole universe of people out there.

We're making sure that we are getting as represented in the sample as we can and make sure we are not just learning the conventional wisdom that we are going to be able to.

[0:53:30.9] JM: So there is a ton of questions that I had for you that I'm not sure we are going to get to like some things around infrastructure, databases, build versus buy, your choice of machine learning frameworks. I want to ask you some stuff about hiring and maybe we can get to that if we do have time but I don't think we do. I wanted to make sure that I got to a bit of a discussion about the future of finance. So you've got this traditional banking industry and credit card industry.

That feels so legacy at this point and then you look at China for example and China leap frogs to the US in terms of payments via phone or via watch or via fingerprint who knows. I am curious where you see the US consumer finance industry going. Is it going to look like what the

Chinese payment system looks like today in a couple of years or will it look like something different? What's your vision for the future?

[0:54:36.2] LM: Well I hope it actually doesn't look too much like the Chinese where they seem to be going with things like a social credit score and using that tool sort of power underwriting that doesn't seem perfectly compelling where who your friend are is driving, you know your assisted credit. In some ways I think it's almost a using technology to go back to basics towards simplification. I mean at the end of the day, people want to manage their money in a responsible way.

They want to have access to credit for large purchases, they want it to be clear and understandable. When they save money, they want to be building towards something and it seems a little like we have a lot of really large companies in the consumer finance space that benefit extremely well from the raw increase in complexity and velocity with complexity that doesn't seem to need to be there. At the end of the day, the users want to understand their finances.

And so I think a lot of where we think about taking it is really using technology. In some ways, I like to say being the first technology company in finance where technology is the actual differentiator of building scale efficiency for customers and customer acquisition and thinking about, "Okay how do you use technology to achieve those goals to provide clarity to the user across all of their finances?" A lot them have access to those finances anywhere they are.

To facilitate what they're ultimately actually trying to accomplish, right? Finances is a means to what they are trying to do not the end goal itself and so I think about almost from mostly from that perspective of how does technology at scale simplify this interaction.

[0:56:29.4] JM: Are traditional retail banking industries, are they adapting to all of these new competition from the firms and the transfer wises and the stripes of the world?

[0:56:41.2] LM: I think they will. Yes and no. I mean obviously it is a very wide set of players in that market that is obviously very large. You have the full range of responses. I always think of the analogy as there's analogy to sort of where Amazon was 15 years ago and as they took on

retail, what does that look like obviously when it is more convenient and simplified. There is this element of obviously there is people in banking that they don't sort of see the issues with where they are coming from.

And how they approach this problem and those are the ones that are going to be the low hanging fruit from our perspective so to speak and there is obviously ones that are extremely sophisticated and having and are going to are not going to just continue to sort of double down on a changing world. They are going to adapt it if it is well. So you have that full range and I think right in the end, you will see a different landscape 20 years out and some of it will be brand new players.

And people who are pushing the boundaries there like us and then you'll have some people who are there today and will be there in 20 years because they understood what these changes meant for them and adapted.

[0:58:10.0] JM: Libor, thanks for coming on Software Engineering Daily. It's been great talking.

[0:58:12.3] LM: Awesome, thank you very much.

[END OF INTERVIEW]

[0:58:17.3] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plugins, use the value stream map to visualize your end to end workflow and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on the fly. GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team who have talked about building the product in previous episodes of Software Engineering Daily.

And it is great to see the continued progress on GoCD with the new Kubernetes integrations. You can check it out for yourself at gocd.org/sedaily and thank you so much to ThoughtWorks for being a longtime sponsor of Software Engineering Daily. We are proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]