EPISODE 1348

[INTRODUCTION]

[00:00:00] KP: Tedious, repetitive tasks are better handled by machines. Unless such tasks truly require human intelligence, they're often good candidates for automation. Implementing process automation can be challenging and technical. Increasingly, engineers are seeking out tools and platforms to facilitate faster, more reliable automation. In this episode, I talk to Yaseer Sheriff, co-founder and CEO of Axiom about no code solutions, process automation, and some of the challenges in developing the software powering those services.

[INTERVIEW]

[00:00:39] KP: Yasser, welcome back to Software Engineering Daily Show.

[00:00:41] YS: Thanks, it's great to be back, Kyle.

[00:00:44] KP: So, for listeners who may have spent a few other things in between your first appearance, can you give us a quick reminder of what is axiom.ai?

[00:00:55] YS: Yes, Axiom is a no code browser automation. It's a bit like an Excel macro where you record clicks and keyboard events only. We do that for the whole web.

[00:01:07] KP: And what's a typical use case?

[00:01:11] YS: So, most people are automating moving data around. Have you heard the phrase ETL? Extract transform load?

[00:01:19] KP: Absolutely, yeah.

[00:01:20] YS: So, it's it's very common to copy data from one system and paste it into another. Now, that might sound a fairly dull but there's a lot of that kind of stuff out there. A very typical use case, for axiom is literally just moving data between systems. It's a very common thing like

you might do that in E-commerce, for example, when you fulfill an order, you might copy it from your your online shop into your fulfillment partners website, for example.

[00:01:47] KP: Sometimes I'll have the option to achieve a goal like that with some sort of software integration like an API or just some connector. Why do I need axiom or are there reasons on top of that, for places that don't have APIs?

[00:02:00] YS: Well, yes, that latter reason is pretty much the main reason, but it's not the only one. So, there's a whole – I'd argue the majority of the web actually doesn't have API's, the vast majority of systems. So, for example, going back to the E-commerce example, if you have lots and lots of different products, and lots and lots of different fulfillment partners, not every one of those will necessarily have an API into the system, particularly if you've got some obscure product. So, you still got a lot of people doing this kind of admin task. There's another scenario where actually like, if you're doing some work, like repetitive work, and it's a very visual thing. For example, producing a design on Canva, or Figma, it's very natural to describe your steps as user interface actions rather than what they are script. It's more intuitive for a person to do that. So, you can say one thing is API and the other thing is, user interfaces might be more intuitive for certain domains like visual ones.

[00:02:59] KP: Would it be fair to say you're kind of creating the APIs on behalf of all those web apps that have not delivered on one yet? Or is that too cynical a way to look at it?

[00:03:09] YS: That's definitely one way to look at it. On a macro view, basically, what we aim to do, and I think what a lot of web automation tools aim to do, because there's a few out there with different spins. We all aim to turn the whole web into an API. It just isn't there yet. So, you could say, we're increasing the total addressable automation space for things on the web.

[00:03:32] KP: So, your solution is no code. I think most listeners will know it, but it might be worth doing a quick definition of no code. But what I'm really curious about is, why is that the right choice for a tool like yours?

[00:03:43] YS: Yeah, so what we do has a whole bunch of developer tools, actually. So, what we do is often called robotic process automation. It's a bit of a misnomer, because there's no

robots involved. But there are software robots and there's a whole bunch of like developer tools for this kind of thing. A very famous ones called UiPath, who I think one of the first or probably one of the biggest IPOs this year, for this kind of space. Those tools are very much developer tools and it means that there's a whole bunch of stuff that your average admin person, or even someone who's doing something high value, like sales, for example, aren't really amenable to automation as you get in touch with a developer.

So, by making it no code, you basically enable a longer tail of things to be automated stuff that okay, maybe isn't something that you can just involve a development team with a large budget to, like a smaller automation that maybe a sales staff or something that changes quickly, that wouldn't be a big development project.

[00:04:44] KP: No code has been quite a trend and I guess a buzzword at this point over the last, I don't know, five years is maybe the time horizon. I'm curious if you have any thoughts on how far we've come as a field that is, you know, developing no code options and where things are headed.

[00:05:02] YS: Yeah. So, this is one of them really interesting things I probably didn't get a chance to talk about last time, but the space is changing so rapidly, we can be really cynical about it. I definitely am, a lot of the time. When I go on Product Hunt, for example, there's another no code tool every other day. So, the cynical take on this is they're all doing the same thing, right? But actually, they're not. What we're actually seeing is this halo effect from all the no code tools coming together. If you're a programmer, you think in terms of programming language, and domain specific languages, and what's actually occurring with no code is you've got this plethora of different programming paradigms, effectively, different domain specific languages to do different things. And when you put them together, you get something a lot more than the sum of their parts.

This is something that we really see with axiom, actually, it's quite exciting to see, someone could use a web automation like Axiom with Zapier or API automations. And then connect that up to something like an Airtable, which is like a no code way to deal with data. One of the really exciting things is actually to see all of the stuff come together to make something bigger. I could

talk about this for quite a while, because we've seen a lot of different movements in the community here. But yeah, it's changing very rapidly.

[00:06:15] KP: No code tends to be fairly democratizing, because anybody can use it, or in theory, I guess, at least a little bit lower barrier to entry, then learning to be a programmer, learning the libraries getting into a code base. Yet some of your services are the types of things developers would probably want to be doing. Do you have a vision or an idea of who your common users are?

[00:06:37] YS: Yeah, that's a really good question, actually. So, a lot of our expectations have been upturned a little recently. We originally assumed that the only people that would use this would be non-technical people and we started with that in mind. What's very interesting with a lot of no code tools is a lot of developers are actually turned into power users of the system. We have people that use Axiom that could write something in selenium, which is the dominant framework, or puppeteer, which is what we're based on as well. But it takes ages to do that kind of thing. Even as a developer, it's just faster sometimes to use a no code tool like us.

I mean, I see something similar with web flow, for example. I can code up a website really easily. But if I had to throw up a little marketing website, I would just use web flow, it's just faster. So, like, what's been really interesting is to see that actually, it's not just non-technical people that use tools like us, it's even developers, when they just want to get a job done quickly, or want to get a job done that they can then hand on to our people in their team to manage. What's exciting about a no code, it's not just non-technical people. I would argue like there's a significant amount of coders and developers who are power users have no code tools.

[00:07:50] KP: When I look through some of the recipes and examples on your site, they seem to be concise processes, I might want to automate. So, there is the example you gave earlier of E-commerce, you want to get a price off a site. Or maybe I want to check my competitor sites, see what their pricing is, perhaps store that in a database. Those all seem like straightforward processes to set up an Axiom. It could be standalone. Maybe that's all I need. Is that one simple service from you. Do you see patterns in adoption? Do people solve one problem? That's good enough for their like enterprise scale rollouts that you're participating in?

[00:08:25] YS: Yes, so that's another good question. There are kind of two ways to run our bots, you could say. One is like a micro service where it's just a really small, tiny little process. But then another way is also to glue these things together. The really tricky thing with gluing all of these things together, and it's actually one of things we're working on is, the more complex the process, the higher the probability of failure, right?

If you're in RPA, the space we're in where you automate using user interfaces, rather than APIs. Failure is like the dirty secret of the industry. A lot of RPA bots actually have relatively high failure rates in comparison to everything else. A lot of the technical challenges revolve around basically making them as reliable as possible, which is a lot easier said than done. So, the more little processes you glue together, the higher the probability that the overall process fails. But therein lies kind of the technical challenge that, we aim to solve, and I think everybody else aims to solve. But yeah, to answer your question, you do get very, very large, complex processes being built up and you do get smaller ones. But like with anything in RPA, the more complex the processes, the more challenging is to automate, basically.

[00:09:35] KP: Yeah, there's a lot of potential for failure, not even something that is your fault or even within your control. There could be a third-party service that like my competitor's website I mentioned, I want to crawl on if it goes down, the task is in a fail. How can you call it down website? But of course, then I'm maybe upset with you, not my competitor. I didn't know their site was down. Do you face any challenges on communicating nature of failure and helping users come with good recovery strategies?

[00:10:03] YS: Yeah, there's definitely challenges there. I mean, the most common cause of failure forum for web automations is the website changing. Anybody who's ever written a scraper knows about this. It's actually the main reason why a lot of people want to outsource that kind of problem to a company like us. Because, why would you as a developer want to spend a lot of time writing algorithms to deal with. The algorithms are completely non trivial. What we end up having to do is thing like, do things like fingerprint elements, if the website changes, look at the structure again, and try to refine that. A lot of the time even that is enough. So, the simplest thing is to allow people to reselect it when the page changes.

But yeah, I mean, you definitely face problems communicating that to people that things change. And these things are by nature a little bit harder than API automations. You could say, like, in the fullness of time, it's that kind of problem that we aim to solve. Certainly, we couldn't solve it for website came down. But being insensitive to page changes, being able to deal with all the complexities of the web is, the crux of our technical challenge, really.

[00:11:03] KP: When it comes to web crawling, I think in most cases, if you obey what websites put in their robots, .txt file, and you're pretty kind about your rate limiting, you're not likely to be blocked or anything like that, just to in my anecdotal experience. However, there are some web properties that, go the extra mile or the extra 10 miles in trying to prevent crawling and things like that. Where do you fall in – what's your perspective on that? How does the tool or how do you want the tool to function when there's a site that's potentially kind of aggressive in defending itself?

[00:11:38] YS: Yeah, so basically, there's this tension between what website visitors want to do and the platforms. So, we don't stop people doing anything, unless it's bad or illegal. So, we have stopped cases, when people have been harassing others on social networks. We've unfortunately seen that and we'll put a stop to it. There's other stuff that that does occur that, which is basically breaking the site's terms of service.

With tools like that, it's not actually our core focus, if you want to go in and like pull data out of the system, we give you – we have a toolset that lets you do that. But that's not necessarily our core business. Where we actually aim to be as something that's kind of like win win for the platform that we're automating. So, a good example is the E-commerce one, because if we help people sell more on Amazon, we're on Amazon's good side. In contrast, with LinkedIn, LinkedIn, don't like you taking their data. So, you're constantly against the platform. Although we allow users to use our tool that way, we don't really want to get caught up in an arms race with all of these platforms. That's not what we aim to do. But yeah, it's a very good question. It really boils down to, in our case, focusing on what we aim to be, win win for us, the user in the platform, but it doesn't always work out that way.

[00:12:52] KP: If I were, let's say, an analyst, or someone who wants to do some process automation, and maybe I've got enough free rein within my organization, I can try my own tools

and explore and things like that. If I want to give Axiom a try, what are my steps to getting set up?

[00:13:08] YS: Yeah, it's just going on the website now and installing the Chrome extension. So, right now there's an extra step was removed very shortly as you have to install a desktop app with it. That's to do with various things with Google, basically. Google has a framework that we used to called puppeteer. But very soon, you won't need to do that, because we have a product that lets you run it in the cloud relatively easily. The long story short, it can be as simple as installing a Chrome extension, working off a template, then building a bot yourself if you can.

[00:13:37] KP: So that's an easy install and setup. What's the workflow like and the learning curve? How do I really get going with my process automation?

[00:13:47] YS: A lot of the time the best way to get going with not just us, but with tools like Airtable or Zapier is look at templates other people have made. I use Airtable a bit as well and making my own Airtable or a complicated one takes a while. It can be a bit of a pain. But if I find the closest thing that somebody else has made to what I want to do, and then work off that Airtable template, I can have something quite sophisticated. It’s very similar with axiom. So, you can try and build something yourself, but if you look at what other people have done, and quite say stand on the shoulders of giants because it makes it sound a bit too grandiose. But if you look at other people have done and build on it, you can go a lot further than just if you try and start from scratch. So, that's what we really recommend as a starting point.

[00:14:29] KP: My process when I'm developing software, or well, what it should be is write the test, make the test work. But more often than not, I find it's, write some bad code, watch it fail and fix it. I just kind of iterate until there's no more things to fix. What's the experience like developing from scratch, if I don't have a good recipe to follow? How do I fail fast and get where I'm going?

[00:14:52] YS: It's very similar to coding, to be honest. That's the thing that some people, some people can come to terms quite quickly. Other people find it a little bit of a shock. Making anything like from scratch can involve a bit of trial and iteration. Just to set clear expectations rather than, like you know, sugarcoat it, it can be tough going to get something perfect.

Usually what happens is, you can get 80% of the way they're really quickly. $It's very similar with to check out proof of concept. And that's actually one of the main things people do with Axiom. You get a lot of startups try and make something that just tests a proof of concept out, and you can get 80% way there really quickly. It's that last 10% to 20%, that takes so much longer if you want to try and make something ready for production, for example. That's when things can be painful to be frank. But I think that's the same with anything in software. The first 80% is relatively easy and it's that little bit of refinement to get something perfect, which really has the hard graft. But I think our users that are sort of power users have no code tools know this already.

[00:15:59] KP: When you think about growth, I imagine a lot of different ways that you could be expanding, maybe a richer library of those recipes that use the existing tools and just show them off in different ways. Or maybe it's new components and pieces that you're adding in. What does growth look like for you?

[00:16:18] YS: Yeah, it's basically expanding out from what we see now is the power user use case. So, when we first – what we've identified as the people that are most capable of using Axiom are preexisting users have no code tools. So, what we found is if you're a Zapier user or an Integromat user, you're quite likely to want to automate a lot of stuff already. It's a good leading indicator to use Axiom. We don't have a Zapier app on the store just yet. It's just about to launch. But a very obvious thing for us to do is a move out from Zapier to Airtable to Integromat to this large ecosystem of tools that like, as I mentioned, has a halo effect.

From there, what we really want to do is lower the barrier to entry, because right now, really the only people that get going are existing no code users. But we think by making the tool easier, we can expand out. I mean, the templates are really good way actually, to test this. Quite often, we'll do the same thing Zapier did, which is basically put up a template, which is kind of like an example of what can be done, and kind of speculatively put it up and see if anybody wants it. Sometimes people just come along and bite and then we find there's this market that we didn't know about before. So, you can say the growth playbook, it's been done before, it's quite similar to what Zapier did really.

[00:17:40] KP: When I think about the ETL use cases, they seem kind of straightforward. Get some data from someplace, maybe process it and store it someplace else. But when I think about process automation, it often feels like a harder problem. There are decisions to be made, conditions to be thought about, filtering to do. Can you talk about how that's all possible in the no code world?

[00:18:05] YS: With difficulty. I think you actually hit the nail on the head for a really hard problem here. Quite often, it's like the decision logic, that's the hardest bit. Doing decision logic in no code is just really not easy. What ends up happening is you need to have to use lots of work around and hacks to try and get a complex decision process to work. Luckily, with Axiom, we're a Chrome extension, so we can do something that Zapier can't do, which is to hand back to the end user to make a decision. So, you can have a workflow where, okay, the bot does a certain number of steps, but then you hand back to an end user to act like a human being to make a decision and then hand back to a bot to take carry on. That's what we do now with a lot of things. But it's a bit of a privilege from basically being a Chrome extension and being attached to end users, which isn't something some of these other workflow tools that run in the background have.

[00:19:03] KP: If I think about some type of software, like let's say, a paintbrush or a drawing application. If you asked me to go build one, I feel like I could just get started, I have a pretty good sense of the primitives. There's paint and change the color, add text. There's the sort of intuitions that someone figured out a long time ago, because of course, paint programs are old. No code, workflow, and process automation are new. Do you feel that those same primitives exist already? Or do you have to invent the language that Axiom uses?

[00:19:37] YS: Well, there's a little bit of both now that you mentioned it. What's really interesting is a primitive is the table and spreadsheet structure. So, if you're a coder, you think about lots of different data types. Arguably, data is more important than the algorithm. Data really defines the algorithm. But what we've actually found is, non-coders, they think in terms of spreadsheets. That's like the data primitive and that's basically how a lot of Axiom automations work. So, for example, the starting point is usually a spreadsheet. If you want to, like do ETL, someone will think about it in terms of a spreadsheet structure. So, we do have that primitive.

The more complicated primitives, I think revolve around, I guess, things in the user interface that people haven't done before. That's when we're kind of making up new concepts. There's a few though, again, sort of harks back to programming primitives as well like teaching a no coder, like the concept of a variable is one. In our case, the variable might be like, okay, what you put into the form, and getting them to think in those terms. But I would say, there's actually an analog to – there's usually like an analog to everything in programming in your particular no code paradigm. There's an analogue to something somewhere, like for example, sub routines, concepts like polymorphism, and abstraction, that actually will do come in, but you don't use those words when you're talking to users.

[00:21:09] KP: A no code option is one of the things that appeals to me about it, it's kind of declarative. I draw out my process and as long as I've drawn it correctly, I can just kind of take for granted that the system is going to handle it for me, although that's really just a shifting of the burden now. Axiom has to make sure to fully deliver on that workflow I've described. Do you face any scalability challenges, as you make these processes people build and see them kind of push them out into either bigger crawls or bigger data sets or something along those lines?

[00:21:44] YS: Yeah, I think the main scalability challenge with respect to complexity is, like I mentioned, going back to the failure point, the probability of failure almost tends to one with the number of steps. So, if you have like a two or three step bot, it's pretty reliable. If it turns into a 30-step bot, there's a lot of other problems where the probability of failure is obviously multiplies as you go through the system, right? Do you have to have these problems of like requeuing things, dealing with error cases, that kind of thing.

[00:22:15] KP: Is there anything unique about being a no code platform for your approach to error handling?

[00:22:22] YS: Basically, you have to have to have a whole bunch of workflows around error reporting, and notifying people about errors and stuff like that, which you don't have as much of that in the API space. So, when you connect to API's together like Stripe does, you don't really think about errors too much. You do get the odd thing where like, “Okay, if this payment fails, there are particular reasons you have to work it out.” But we have to start to do things like, when the bot fails under these conditions, send a message to someone. That does introduce

significant challenges. The most common things that we need to, probably need to solve are basically around requeuing when there's problem cases and notifying people in case of errors. Because the way we solve this now is basically to hand back to the end user in some respect. I mean, like I said, the double-edged sword about being in the browser is you're attached to an individual, so you can actually hand back to them.

[00:23:17] KP: When you think about process automation, what industries or maybe specific companies are really doing it well?

[00:23:24] YS: I've mentioned Zapier lots of times, because I think they're really doing it well and Integromat too. They've recently emerged as a competitor. I guess, they're basically like Zapier with different pricing. There's a lot of these automation tools that exists that people don't actually like them. Those are usually like enterprise tools, that you get this phenomenon where a lot of enterprise stuff is just sold via set. It's just like a good sales process and not necessarily a good product. But these bottom up tools like Zapier and Airtable, they're the ones that do it well, because people want to use them. They're adopted bottom up in an organization. The only way Zapier and Airtable succeed is if their product is excellent.

In my opinion, anybody who's doing a bottom up workflow automation in any way, and succeeding must be doing it well, by definition, because people are choosing to use the product, they don't have it imposed on them with enterprise automation stuff.

[00:24:21] KP: Are there any novel or interesting use cases you can share?

[00:24:25] YS: Yeah, there have been lots of weird ones. I mean, okay. The novel ones, I could actually write a whole blog post on this, but I'd have to be careful about what we talk about. Some of the more interesting ones revolve around, I guess, people backing up their private image galleries. You can imagine what people's private image galleries might be like, that seems to be a thing. There seems to be a lot of people who do – people people who aspire to the four-hour workweek who've built a bot to I don't know, do something drop shipping and automate some whole series of obscure stuff. The four-hour workweek persona is someone we talked about a lot. Basically, they've worked, they found some niche thing and they want to automate it.

Some of the more interesting ones actually are around crypto lately. We've started to see a lot of people – because crypto is like a wild west, and you don't have API's and lots of systems. So, we start to see a lot of like day crypto traders, monitor NFT's and things like that. So yeah, I mean, the more interesting ones that anytime you get a new nascent industry, which is like a wild west, sometimes like it's in finance, for example, where you don't have APIs and system, you start to see all these weird things emerge where people will try and automate it just because, like I said, doesn't have an API, and that correlates with anything new, basically.

[00:25:53] KP: The advent of the pandemic has changed a lot about the way people work, were more remote and things like that. Has it changed any of the way people leverage Axiom?

[00:26:02] YS: Well, to begin with, the pandemic has been really interesting, because lots of things have gone back and forth. With people working from home, there seem to be a little more people running like little side hustles here and there. That seems to been a bit of a thing like that. The notion of a side hustle has been a lot more common. There have been some interesting pandemic related ones. I don't know how moral they are. We've had a few people try and book COVID vaccinations during the peak of the vaccination thing. We were wondering whether to block that one. Because it was like, “Well, it's unfair if you're using a bot to book a vaccination.” But we let them do it anyway. That was one of the more interesting ones.

[00:26:41] KP: Yeah, yeah, I see the debate. That's a tough one.

[00:26:44] YS: Yeah, we saw it happening quite regularly. And we were just like, it was kind of like a medical ethics thing. But yeah, I think, that particular phase has passed now, as you know, vaccinations have become more prolific. We see some other ones which are like, we've started to move away from as well, the kind of people who buy sort of arbitrage products, the sort of scalper cases. Initially, we were okay with that, because it seemed to look like E-commerce, but we started to like, really deprecate it.

[00:27:14] KP: So, when it comes to those side hustles, and maybe early stage entrepreneurs, is the idea that Axiom is just one of many tools are used, or is it potentially the platform upon which I build my business?

[00:27:27] YS: Well, actually both. So, by definition, it's often not the only tool. By definition, it's often connecting together other systems. But it is a platform where people build their businesses. So, we've had people build unique businesses on Canva, where they make templates on Mass with Axiom, we've had people build unique business on Webflow. Again, where they make templates on Mass. We've had Fintechs use Axiom when it's basically how like – I mentioned this the last podcast, but like, how our plan got started when you don't have APIs into banks.

When they started doing this, we found it a little bit concerning. People would like automate online banking stuff to basically simulate an API because they had no other option. But yeah, that's been an interesting one. So, it's really common with startups, actually. By definition, startups are trying to do something new and very often new things don't have APIs. So, we really do see, we see lots of people try and build the businesses on us. Food delivery seems to be another one that seems to be cropping up. Lots of places don't have, like their menus as APIs. So, it's a prolific place for bots, basically.

[00:28:41] KP: When you look at the biggest adopters, people who are doing the most with Axiom, what backgrounds are they coming from?

[00:28:48] YS: Startups and SMEs seem to be the majority ones or solopreneurs. When we first started, actually, we were looking at the enterprise sector, but the way this kind of automation works, the enterprise is top down. You do it by enterprise sales and all these other things. We both Axiom, like I mentioned as a bottom up product, and it's very much in the mold of Zapier and Airtable. If you look at how Zapier's trajectory has evolved, it pretty much – I think now it's used more in enterprise, but certainly when it started the early adoption was an SME and startups. Our typical user is in SME startup, and or solopreneur. A time for solopreneur is the kind of person who loves automation. They've used a no code tool before. They may have some existing workflow setup and they see how Axiom can maybe plug the gaps in their Zaps or something similar.

[00:29:41] KP: Can you speculate on what the adoption of process automation looks like, maybe 5 to 10 years out?

[00:29:48] YS: Yeah, I think it's going to be everywhere, really. I think right now what ends up happening is, automation is always a big project. People don't bother to automate because there's like this activation energy, where it takes developer time, it takes a lot of effort, all these other things to automate something. So, things just don't get done that way. A lot of technology, just when you move something forward, you just make something cheaper. What no code is doing is making automation cheaper, right? So, I kind of see as these no code, as these automation tools become more prolific, they lower the activation energy to automations. They kind of make anything, no matter how seemingly trivial or automatable, we can only see that trend increasing.

[00:30:32] KP: So, if I install the Chrome plugin, does that help me with interacting and building workflows and things? Or does that mean the work actually runs on my machine?

[00:30:42] YS: They can run on your physical machine or they could run in our cloud. We had, I forgot to mention some of the COVID ones, we had a bunch of COVID data entry after vaccination use cases are run by doctors on Axiom. They didn't want to run it on our cloud for patient data reasons. But everything ran on their machine. We didn't touch or see any of their data as a result. So, people run both modes, depending on what they want to do.

[00:31:07] KP: Can you expand more on the cloud offering? Why do people elect that option?

[00:31:12] YS: Yeah, so the cloud is usually for when you want to remove it from what goes on your machine, either as a response to an event like a Zap, or if you just want to run continuously. So, on the price comparison case, we have a bunch of people running bots every minute to get prices on things, and you wouldn't really want to run that on your machine, you don't leave your machine on 24/7. That's a classic cloud use case along with like, let's say, connecting it to a Zap or an Integromat. You have plenty of Zaps for, if this happens, do this.

So, what we sometimes see is, if you get an order come in London, the Airtable and trigger a Zap, and then an Axiom bot spins up and then goes and does something. That has to be in the cloud, really. You're not really going to connect that up to a local machine. That will be something where it's just running 24/7 and ready to react.

[00:32:01] KP: Can you share some details on your roadmap? Where is Axiom headed?

[00:32:06] YS: Yeah, the main thing right now is just to try and reduce the activation energy to go back to that point. So, right now, getting started requires a little bit of a setup process. I would describe it as potentially high friction setup process. But within the next few days, actually, we're launching a new version where people can get up and running without having to do much setup at all. They literally install the Chrome extension, have access to the cloud, and can run things immediately.

The other phase, I guess, is just teaching all of the concepts to non-technical people. We do know, like I mentioned, there's this subset of power user that gets it really easily by a lot of people come in, and if they're not familiar with – normally with variables, for example, if they're not familiar with data structures, and things like that, they can struggle. So, it's really about teaching that to the average person who maybe doesn't think in terms of data structures, and variables and moving stuff around spreadsheets and forms. But if we can teach them that, we kind of really widen the total addressable market for automation in general. But yeah, that's the next phase after this next product update to make it a lot easier for the average person.

[00:33:15] KP: Well, it sounds like a good milestone, and maybe a time for our listeners to consider if this could be a good tool for them. Can you remind us what is the hello world scenario?

[00:33:26] YS: Yeah, the hello world would be like getting data and displaying it to you or writing it to a sheet. It's pretty easy to do that. Go to a – I recommend trying on a very simple website, like let's say Wikipedia, for example, which isn't so fancy. Some of the other more complex websites you see. Go on that, play around with it, select some data. See if you can get it to appear on the page or get it to write to a Google Sheet. It's a good way to get to grips with moving data between systems, which is quite often the core of what we do.

[00:33:55] KP: Where can people learn more online, if they want to follow up?

[00:33:59] YS: Head over to axiom.ai. Let us know if you run into any problems. We've got reasonably responsive support these days. So, I'm happy to help.

[00:34:07] KP: Yaseer, thank you so much for coming on Software Engineering Daily.

[00:34:09] YS: Thanks a lot, Kyle. I appreciate it.

[END]