1:39

Alexey

**This week, we'll talk about working on open source. We have a special guest today, Will. Will is a software engineer and author. He's quite an enthusiastic developer of open source software and he is the creator of some very popular Python packages. Welcome, Will.**

1:54

Will

Thank you. It's good to be here.

# Will’s background

1:57

Alexey

**Yeah. Before we go into our main topic of working on open source, let's start with your background. Can you tell us about your career journey so far?**

2:07

Will

Career journey so far – condensed 25 years and a few censors. Well, I started out in video games. I was working in video games for about 10+ years: PC games, PlayStation games, even Dreamcast. Then I moved into desktop applications. I worked for a while for the Internet chess club. Then I worked in various new media websites, that kind of thing. Prior to my current position I was freelancing, doing mostly Python stuff, mostly web-related – building servers and protocols. And very recently, I've started a company called Textualize.

2:57

Alexey

**Yeah, that was pretty condensed. 25 years you said, right?**

3:01

Will

Yeah. [chuckles]

3:03

Alexey

**For your work at the chess club, did you have to play chess well?**

3:10

Will

No, I'm a very weak player, to be honest. I'm better at implementing software than I am at playing chess. And I was working on a new interface for the Internet chess club. It was like a chess board where you can move pieces around and draw arrows and exchange chat with other players.

3:30

Alexey

**[chuckles] I was imagining an interview where you need to beat a couple of people in chess in order to pass the interview. It wasn’t like that, was it? [chuckles]**

3:37

Will

I mean, I quite like chess. I did play a little bit. But the reason I got that job was because I built some chess software in my spare time a few years previously, and I was selling that. This is before my open source adventures. This was something to make profit from. Well, I didn't make a lot of money, but it brought in a few thousand pounds. But it got the attention of an internet chess club, so I got a job. So it's not so bad.

# Will’s open source projects

4:10

Alexey

**Yeah, cool. So speaking about open source, can you tell us about your open source projects? What are they? What did you work on?**

4:18

Will

I had a few small projects a while ago. I had a BBCode parser. I don’t know if you're familiar with BBCode.

4:28

Alexey

**This is something we used for old PHP forums, right?**

4:34

Will

Yeah, it’s for bulletin boards. If you want to add, or type something that had bold in it, you would do [B].

4:42

Alexey

**Yeah, it’s like HTML, but with square brackets.**

4:45

Will

Yeah. And at the time, I was adding that to a website and there wasn't a BB code parser, so I built one for Python. It started to get a bit of traction, because at the time other people in the same session wanted to implement BB code. So I think that's my first… Actually it’s not my first, but the first project which got traction and people started using. I'd written other things prior to that.

5:15

Alexey

**Was it something that you just saw that you really needed and thought, “Okay, there is nothing like this, so let me just implement this.”?**

5:22

Will

It’s exactly like that. Yeah. I just needed it for a website. I Googled and couldn't find anything that did it. So I just implemented it. It was quite interesting. It's a simple kind of parser and render. Most of my open source stuff has been like that – I want something to exist. If it was something that someone already did, I would have just taken it off the shelf. But when it doesn't, I think about building it myself.

5:52

Alexey

**Sorry, I interrupted you. You wanted to tell us about the next project you worked on after the BB parser.**

6:00

Will

Well, there's a few other small projects, which I'd open sourced. What was it? Well, there was my chess library – a library which I wrote over a few weeks, which parsed chess moves, read PGn files (which is what stores chess game information) and I published that. It was quite buggy, actually. There were a few bugs. But people submitted PRs. I'm not sure if they were called PRs at the time. [chuckles]

6:34

Alexey

**Was it on GitHub. Did GitHub exist back then?**

6:38

Will

This was pre-GitHub, I'm pretty sure. I think it might be…

6:41

Alexey

**What was it?**

6:42

Will

Google code.

6:43

Alexey

**Yeah, remember this thing, too. But they closed it off. [cross-talk] Maybe good for GitHub, but I kept quite a few projects there.**

6:52

Will

Yeah, it wasn't bad for the day. It used SVN. I quite liked SVN, but GitHub took over. Yeah, so I had a few projects on there. My next project that took off was something called the PyFile system. Again, it's a Python library which abstracts file systems. A file system would be like your hard drive, but it could also be an FTP server, or it could be a zipped file, and the PyFile system would provide a common interface onto all those things. You could write some code, which by default, would read files off your hard drive. But if you wanted to read files by cloud, you could do that by just swapping one line. Or if you want to write a zip file, it's just a matter of copying it between two file systems – that kind of thing. So it was kind of an abstraction layer for file systems.

7:52

Alexey

**I think it's pretty commonly used still, right?**

7:56

Will

It's still quite popular. It's been, you know – it was over 10 years ago – and it's on version two of the interface. And it's still being used. I had to give it over to other maintainers because I just didn't have the time to maintain it. That's the tragedy of open source. Once you build something that gains traction, there's no end. [laughs] There's no end to it. People use it. They will want fixes, they want updates and refinements. And I did that for nearly 10 years. But at some point, you have to say, “Sorry, I have to go into another project.” But fortunately, there's a couple of very talented developers who are already submitting PRs and who are already maintainers. I just had to leave it to them and they've done a fantastic job so far.

8:54

Alexey

**That is also the beauty of open source. You said it's a tragedy, but it’s also the beauty, right? Because it’s there for people because it is open. So if you say, “Okay, this is enough for me.” The code is open and anyone can come in and keep working on this.**

9:12

Will

That's right. And that's the awesome thing about open source. Your little pet project can become something bigger, and other people can fairly naturally migrate to it and start producing content and fixes. So a project can be bigger than the one person that started it.

# S3Fs and PyFile systems

9:35

Alexey

**And S3Fs is also your thing?**

9:40

Will

Yes, S3Fs is a PyFile system, so it has the exact same interface as all the other ones, but it writes to Amazon S3. The tricky part was that Amazon S3 isn't a full file system. It looks *kinda* like a file system.

10:03

Alexey

**Like the folders are fake, right?**

10:05

Will

The folders are fake. There are basically just files and, by convention, you can impose a kind of a logical file system. But it's not a full file system. But the interface is flexible enough in that it can present the illusion to the developer that it’s just a file system, like your hard drive.

10:30

Alexey

**Just a few days ago, I discovered that in Pandas, you can read and write to S3. Then, of course, you know what it uses for reading and writing to S3?**

10:44

Will

Does it use S3Fs?

10:46

Alexey

**Yeah, they do. Yes. They made me install it, it says “Okay, it looks like you're trying to write to S3, but you need to install this dependency to be able to do this.”**

10:56

Will

Okay. Is it the PyFile System version? Because I think there's another package called S3Fs.

11:03

Alexey

**I don't know. So I just needed to do pip and install S3Fs. I don’t know which version exactly it was.**

11:10

Will

I think that's another one. I think my S3Fs on PyPI is fs\_s3fs because they came before, for me. I was very unimaginative regarding package names back then.

# Inspiration for open source projects

11:29

Alexey

**I see. [chuckles] I guess I already answered for you regarding how you come up with these ideas. So you see a need – you need to do something, for example, you need a BBCode parser, or you need to have a way to access files, like a common way of working with zip files, FTP files, S3. There is nothing available so you go ahead and implement this. Is this how it works for all your open source projects?**

11:55

Will

Often. I think the PyFile system had an interesting origin. That came from when I was working for the Internet chess club. They had a chest interface, which came with a bunch of files. We were implementing a plug-in system where you could distribute a zip file. And that would overlay the existing file system with other files, but do it virtually. It didn't copy over files, it just kind of transparently presented the illusion to the application that it was reading combined file systems.

So I made something *like* the PyFile system for that. But when I left that chess club, I realized that this is quite a flexible system, just creating this virtual file system illusion – you could put anything behind it. So I implemented that in my spare time. That took him a few months to come together. But that’s how that started.

13:03

Alexey

**So there was an idea that you got inspiration from – from your full time job – and you thought, “Okay, it doesn't seem like there is anything like that out there.” And you thought, “Okay, let's just implement this.”**

13:15

Will

Exactly, yeah. That's how most of my projects start. Sometimes it's intellectual curiosity. Like, I don't know how to implement something – I've got some big ideas. Sometimes the best way to learn is to give it a go and see what you can come up with. I've got *way* more abandoned projects, then projects that become popular.

13:43

Alexey

**Do you publish everything you do in open source? Or are there some things that aren’t? I think you mentioned this chess thing – I don't remember, something related to chess – that you said wasn’t open source, right? So some things that you’ve made are not open source, right? [cross-talk]**

14:01

Will

Yeah, that was open source – on Google Code. But in those days, Google Code didn't have the collaboration features, so it was just kind of something that I did in my spare time, on my own computer. And then I thought “Someone might be interested in this.” And I would upload it to Google Code. Some people did take it and put it into their own projects and fixed bugs there, and some of those got fed back.

14:30

Alexey

**You said the thing that got you a job at the chess website – you mentioned that you were selling something for a couple of hundred pounds per month. Did you already make money with open source or was it a different thing?**

14:44

Will

No, that was something proprietary that I was just working on as a hobby. I want to turn it into a living but I couldn't quite do that. I built a chess interface. I think it was in C… C++. That's how I learned all about parsing chess and validating moves and such.

# Will as a freelancer

15:07

Alexey

**I know we didn't cover all your open source projects. There are some more recent ones, which we will probably talk about later, when we talk about your career or your experience as a founder. But I wanted to go a little bit back in time, maybe 10 years ago. I checked your LinkedIn and I saw that your last full time job was in 2010. And since then, you've been a freelancer. So why did you decide to go solo? Why did you decide to start freelancing back then?**

15:37

Will

It wasn't a particularly conscious decision. I was in a job at the time that I wasn't enjoying. I was looking on, I think it was a Python job board, and there was someone looking for a Python developer to do three months’ work. And I thought, “Well, that's quite good. I can leave this job that I don't enjoy. I can do this few months’ work, and the pay was quite good. That will give me a buffer to find my next full-time position.” Anyway, those three months turned into another six months. And that contract just kept being extended. I was with that company for more than 10 years – 11 years, actually.

16:23

Alexey

**Ah, so it's like, not quite freelancing. It's freelancing, but not quite, right? [chuckles] Because it's the same company.**

16:30

Will

Technically freelancing, yeah. For tax purposes.

16:38

Alexey

**Interesting. So it can also work like this. I don't think I’ve ever met anyone who's done anything like that. Actually, it's interesting in Germany – if you try to pull off something like this, the tax authority will come after you though. They will say, “Oh, something is suspicious here. It looks like full-time employment. Who are you trying to cheat?” [chuckles]**

17:08

Will

Yeah, we have that here as well. You have to be very aware of the rules. But we stayed on the right side. So it was technically. I actually did other jobs as well. There were a few jobs that came up – much smaller contracts, generally. But the fact that you can do two contracts at the same time is indicative of freelancing, because with a full-time position, you're generally not allowed to take on any other work. So, yeah it was a bit of a conscious effort to stay on the right side of the rules.

17:48

Alexey

**This experience of being a freelancer, working with multiple clients (long-term clients and shorter contracts) was it something that also gave you inspiration for your open source projects? Or were your open sourcing activities quite unrelated to your freelancing?**

18:11

Will

Strangely, the open source stuff was kind of like an escape from the contracting. Because with the contracting you've got lots of business requirements and you have to integrate with other people's software. You have much less scope, or at least I found that in that position I'd had much less scope, to make the big decisions and to influence how the project grows. So what I found was that I would do a project on the side. When I did that, I was like, ‘master of my own domain,’ you know? I can make all the big decisions and influence the architecture. I found that bizarrely to be kind of cathartic – a kind of a release from my day job.

19:05

Alexey

**It’s interesting – many people who work in IT, developers, also have development of software engineering as a hobby. To escape from their day job. [laughs] [cross-talk]**

19:15

Will

That is very strange. It'd be like a doctor who did appendectomies at the weekend, just to relax. [chuckles] It kind of doesn't make much sense.

# Starting a company from a tweet (Rich and Textual)

19:26

Alexey

**Yeah, that's interesting. Maybe let's talk a bit about your more recent open source projects. I know that one of the projects, which you will tell us about now, actually led to starting a company. Can you tell us how it happened? What's the story there?**

19:49

Will

Yeah, sure. So I think the project you're alluding to is Rich, which is another Python library, which can write interests to the terminal. You've got like color and style and formatting and tables and things like progress bars and spinners, etc. It started out with a much smaller vision. I had this idea. Several years ago, I was working on a web framework – again, another hobby project – and as part of that web framework, there was a module called console.py. And console.py had many of the same ideas as Rich, in that you could write a kind of a markup, which would insert color and style into your text. And it could manipulate that text to produce more interesting formatting.

When I left that project, I still had in the back of my mind that maybe I should take that console.py and factor it out into another library, because it was quite useful. I found myself wishing that existed when I was working on other projects. A few years have passed and then I had some spare time when I was traveling and I thought, “I'll give this a go. I'll just… I'll write something else. I'll figure out how to do these anti-codes to add color.” Then in a week or two, I had my “Hello world,” or my “Hello world” was in bold magenta in the terminal. [chuckles] That was the start of it. I built a flexible system for applying style to text and still allowing the developer to format that text.

21:40

Alexey

**And all this was in terminal, right?**

21:42

Will

All this was in the terminal, yeah. There were lots of libraries which did similar things, but what I found was that they didn't integrate very well. So if you had a library which, say, turned some text into bold or italic, or gave it some color, that would work. But then you've got another library, which does tables and you want to put some formatted text inside the table style – and you couldn't. Once you format the text, you couldn't manipulate it afterwards. So there are all these libraries, which did a fairly good job of what they were intended to do, but it didn't work together. You still couldn't do a great deal.

I think you could see that from the projects in those days, that when it did add color, it tended to be a single line at a time. Like, if you saw a table, it tended to be just two colors – just foreground and background. So I wanted to build something which took all these ideas and put them together in a way where you could functionally integrate them. If you have some tags, you could put it in a cell and it would wrap properly. And you can also style that text in advance, so you could apply, say, syntax highlighting to code, put that within a table cell, and that will just render as you'd want it to. I built this system, and then I published it and it took off, basically. I think other people found that need. I started getting lots of…

23:21

Alexey

**Which is quite interesting. Usually, developers who are in the terminal world – who are into CLI (command line interfaces) and I'm one of those people – I’ve found that maybe these people care less about visual aesthetics. But they care more about the functionality, right?**

23:48

Will

Yeah. And there’s not necessarily a conflict there. If you use Rich, it doesn't break any of the functionality of a terminal. For instance, if you…

24:00

Alexey

**It just makes it nicer, right?**

24:01

Will

It makes it nicer, yeah. But if you pipe some of the content that you've done with Rich into another application, it will strip off the color. So it doesn't break anything as far as the terminal apps go. I think even those people that said, “Oh, I don't mind. I just want to see the content.” When you have a bit of color and style, just emphasizing strings, or formatting data structures, they go, “Okay. Yeah, this – I actually wanted.”

24:36

Alexey

**I must admit that the CLI tools that I use that have these features, they're more interactive. It's not like that actually matters at the end – when it comes to functionality – if it gets the job done, it gets the job done. But still, there is this appealing element that is quite nice. This little touch that’s like, “Okay, now it's just more pleasant to look at.”**

25:03

Will

Yeah, more pleasant to look at, but we say something is more pleasant, it also means it's more readable. You can pick out the information that is more relevant quicker. And if all your command line applications are like that, it saves you time. If you've got pages of output and you're scrolling through, and it's quite dense – it takes time for you to visually parse and pick out the information you need. But if that’s formatted.

25:37

Alexey

**[cross-talk] logs.**

25:37

Will

Usually it’s logs, yeah. And Rich has a log formatter, which puts things in nice, neat columns and it colorizes appropriate bits of information. So you can pick out…

25:49

Alexey

**Can you just pipe it? Let's say I have a boring, dull log from, I don't know, my Python server. Then can I just pipe it through Rich to make it more pleasant?**

26:03

Will

Not by default, no. But Python has a logging library and Rich has a handler for this. So you can swap out regular Python logging for Rich logging. And then you get nice, neat, formatted output.

26:21

Alexey

**But I guess when you write it to a file, it's all lost. Right?**

26:25

Will

It's all lost, yeah. You won't see it when it's in the terminal.

26:32

Alexey

**Okay. So you’ve worked on this library, Rich, and what happened after that?**

26:39

Will

So Rich was becoming really popular. At some point a couple of guys, very tolerant guys, they took Rich and they built a text user interface with it, which would display information from the GitHub API, which are like recent pull requests, and it divided the screen up into four and showed like scrolling windows, and it formatted it really nicely.

When I saw that, I realized that there was so much potential to take Rich and then build a framework around it, which did more dynamic things. So instead of just writing content to the scroll buffer that you could scroll through, like most applications do, it could take over the entire screen. And it could present [cross-talk]

27:33

Alexey

**[inaudible] stop, right?**

27:34

Will

Like a stop, yeah, but that's the two E's of yesterday. You can do so much more. Terminal supports 16.7 million colors. It's very rare to see applications that actually use that. And you can update things quite quickly.

27:56

Alexey

**[cross-talk] the entire webpage there, right?**

28:01

Will

Sorry?

28:02

Alexey

**You can have a web page there – you said it supports 16 million, like basically the whole RGB space, right?**

28:08

Will

Yeah. You can create something which works much more like a web page. I mean, it's obviously very limited in what you can present, but it can be remarkably functional. And these applications, if you're familiar with a web page or a desktop application, even if you're not a technical user or a developer, we can sit you down in front of these applications and you can use it because it's got lots of familiar controls. Anyway, what I just described there was Textual, which is a framework that I started building – which is built on top of Rich. I was really pleased with where that was going. I realized there’s a lot of potential there and I had this idea for a business.

I was at a point in my career where I thought “I should take a year off, live off savings for a year, and give this a go.” Because I thought it was a great idea. “At the end of that year, maybe I could approach someone with venture capital to build a company around it.” So I started blogging about this – sorry, not blogging, Tweeting. I’m kind of an avid Twitterer. And I think it was Tweets, which got the attention of a venture capital firm and they approached me. At the time, I hadn't fully explained my idea for a business, but I discussed it with them. It took a while, lots of backwards and forwards. I refined a business plan of sorts, and then I got investment. So I was unemployed for barely three months.

29:59

Alexey

**So you gave yourself a year, but after three months you got investment.**

30:03

Will

Yeah. I thought that I would need to have something approaching a finished product before I could go to someone and ask for money. But the venture capital scene at that time was more – what’s the term? Bearish or bullish, I can't remember. But they were willing to invest in a good idea, something which had potential. So we got, what I think is called pre-seed funding. And it was enough to start a company and employ some helpers.

30:40

Alexey

**What did this tweet look like? I imagine you didn't specifically write “Hey, this fixes this bug. By the way, if you're a VC, give us some money.” It didn't look like this, right?**

30:51

Will

[cross-talk] No, not at all. Because my plan was to take that year off and *that's* what I was tweeting about. I think that just got the attention of the VC company and they approached me. Certainly in our first conversation, I thought it was just someone wanting to discuss what he's working on. Because I do that – I like to talk to people in the Python community and exchange ideas and go over the projects they’re working on. But then I soon realized that, “Okay, maybe this is an opportunity to get investment.” And certainly, that was much more attractive than me burning up a year's worth of savings.

# Building in public (Will’s approach to social media)

31:40

Alexey

**Of course. I think this approach is called “Building in Public,” right? When you just do something – work on your project – and then every little thing you work on, you share it on social media (on Twitter). Right?**

31:53

Will

Yeah, pretty much. I mean, you do that without any expectation that anyone's going to read it or see it. But if there's people out there that might find that interesting, and they start to follow you. The thing about the stuff I was working on – Rich and Textual – is it's by its very nature, it's quite visual. So I can work on something and then I can post a screenshot or a video.

32:19

Alexey

**“This is what I did today,” right? “This is the bug I fixed.” And then you just take a screenshot and post it.**

32:24

Will

Exactly, yeah. “Here's the feature that I'm working on,” and it can show progress. So initially, the first few hours’ work, I get something that's very rudimentary, but it kind of looks like it might turn into something nice. Then every time I add a feature or refine it, I can post a new video, and a new tweet, explaining what I did. And people follow that, because it can be quite interesting to some people.

32:50

Alexey

**Did you do this with the intention of getting somebody's attention? Or was it out of habit? Why did you actually tweet about this?**

33:01

Will

I guess it's just something that *I'm* interested in. And that's what Twitter is for you just kind of… [cross-talk]

33:08

Alexey

**Like, “This is what I ate today.” [chuckles]**

33:11

Will

Well, I sometimes post pictures of what I cooked, but no one was interested in that at all. I did notice that people did actually react to the screenshots that I posted. So that kind of motivated me to do more of that kind of content.

33:28

Alexey

**Yeah, the comment about “what I ate” – I started my Twitter journey in 2020. Before that, I thought that Twitter is the place where people share the food they eat, like “This is the burger I ate today at McDonald's. This is my french fries.” And I thought that this is what Twitter was for. But I was so surprised when I learned that it's actually not that. Now it's Instagram where people do this kind of stuff. [chuckles]**

33:56

Will

Well, it still is that a bit – and more. There's lots of bubbles. The people that follow you and the people you follow, they create kind of a bubble. You get people that are like-minded, who are interested in the same things that you're interested in. I mean, Twitter can be a really horrible place. [chuckles] Don't put anything political. It's a bad idea. But if you get that bubble of people that are interested in the same thing you're interested in, you can get a lot of feedback from what you do. And you can learn a lot from them, and it can be very positive.

34:34

Alexey

**Now, I think the last time I checked, you had quite a few followers, right? It was 18k or something like this?**

34:42

Will

Yeah. Eighteen thousand. Crazy.

34:45

Alexey

**Was it like that when you got approached by the VC?**

34:50

Will

Um, yeah. Slightly less, obviously. But yeah, I did have a number of followers. I think it might have been like 12,000 at that time.

34:59

Alexey

**So you had this habit of just tweeting about what you work on. And then over time, you accumulated a decent number of followers. Then, I guess, when you have quite a large – I don’t know if “audience” is the right word – but people who follow you, then it's natural that you get attention from people like this VC. Right? But it wasn't your intention.**

35:27

Will

It wasn't my intention. [cross-talk] At the start, certainly. I mean, later on – I won't lie, I have sort of gamed the system. I try to get people to react to it, I try to post things which people might be interested in. But initially, it was just me. Because software development can be kind of a solitary thing. I mean, I'm married, but my wife is not in the same field – she's not a developer and she doesn't get excited about the same things that I get excited about, as far as software development goes. So it's quite nice to be able to reach other people that are interested in what you're interested in. That's why I did it. I just put something out there and got a feel for what the reaction was. There was never any intention of trying to attract investment.

36:26

Alexey

**So I guess one day you woke up, and you saw a direct message that said, “Hey, take our money,” Right? And then you thought, “Okay, let's take it.” And you started the company. Right?**

36:39

Will

It wasn't quite as direct. [laughs] I mean, obviously, they want to do the due diligence. They'll discuss the business plan with you and they want details. There's lots of backwards and forwards. But no, honestly, it didn't take all that much time. Everything I've read told me that when you come to look for investment, it will take you a long time – you’ll have to approach many investors, and you'll have to get used to getting knocked back and refining your pitch. But for whatever reason, that wasn't the case. For me, it was actually a little bit easier.

# The workforce and roadmap of Textualize.io

37:18

Alexey

**Okay. So you started the company. So I guess you're the CEO of the company, right?**

37:24

Will

Yeah, that's right.

37:26

Alexey

**And you said that you got this pre-seed round, which gave you enough money to actually start hiring. Right?**

37:32

Will

Yeah. I've got two other developers now. We will be hiring again, probably within a few months.

37:40

Alexey

**Okay. So they gave you some money with the intention that you spend this money on hiring developers, right?**

37:50

Will

Well, just running the business. But developers are generally the most expensive part.

37:55

Alexey

**Basically, they just gave you the money and said, “Do whatever you think is necessary,” and then it's up to you to decide if you want to spend this money on developers, on marketing, or on whatever?**

38:08

Will

Yeah, essentially. I mean, you still have a controlling share of the company, so it's your company. But, you know, they want you to succeed and they know the tech industry. So they're good to have on your side. They can help you navigate the world of tech startups.

38:32

Alexey

**And how do you earn money with Textualize? *Do* you earn money with this?**

38:38

Will

Um, no. Not at the moment, we're in kind of a research and development phase. We're building textual – creating the framework, which you can use to build applications within the terminal. We’re trying to make that as elegant and as beautiful as we possibly can. Then what we want to do is build a kind of web interface, which takes those terminal-based applications and turns them into web applications with a single switch. So you can build an application, distribute it on PyPI, or the usual channels, if you wish. But you can also build it and then serve it on the web just incredibly easily.

That will make it available to non-technical people, as terminal applications are generally kind of a walled garden to almost exclusively developers, and some other technical people that use them. But a textual application can be usable by non-technical people. I think there's a big market there. It’s probably a very small sliver of the web application market, but I think if we do it right, we would have the whole sliver to ourselves. If that makes sense.

40:01

Alexey

**Yeah. Because VCs don't give you money just to have fun, right? They want to see that eventually, in a couple of years, or I don't know how long, that you will be able to return this money – to give them a return on investment. So they need some sort of business plan for you, like “In two years, we want to earn money *this* way,” right? So phase one is nothing concrete. Or how concrete should it be?**

40:28

Will

Well, there has to be a roadmap. I mean, sometimes when you're in this kind of research and development phase, the exact nature of how you're gonna make money might not be that clear. But there has to be a roadmap, where you can see that you could have a viable product, which people might want to pay for in the future. The thing I like about what we've done is that it can still stay true to its roots as open source. I'm still building something which is open source – you get *all* the code and you can do what you want with it.

But this web service is an add-on thing. It doesn't take anything away from what you got, it just gives you an extra feature. We do plan on having a very generous free tier for individual developers, open source projects, and it'll be free and you can serve it to quite a lot of users. So beyond that, we would charge for some features for more enterprise-type of uses.

41:33

Alexey

**I think there is an app called Streamlit – maybe you’ve heard about this…**

41:37

Will

Yeah, they were recently acquired for 800.

41:41

Alexey

**Yeah, that was a good, good number. [chuckles] But the business model they have is somewhat similar, right? They don't have a terminal GUI, but the business model itself – you can cost a few things for free, but if you want to go beyond this limit, then you have to pay something.**

42:03

Will

Yeah, it's quite a similar business model. They even have a Python API, and we will have a Python API. So it is very related. Their audience is probably slightly different from ours. We are aiming for more engineer-type use cases, engineer tools, developer tools – something to bridge that gap between engineers and the managers and their bosses. But Streamlit is data science/machine learning. But yeah, the model is quite similar.

42:40

Alexey

**And I guess you wouldn't mind being acquired by Snowflake. [chuckles]**

42:43

Will

I wouldn't mind. If they’re listening. [chuckles]

42:47

Alexey

**[laughs] Yeah. If you're from Snowflake, please DM Will. Anyways. You mentioned hiring – you already hired somebody?**

42:59

Will

Yeah, we’ve got two developers. Yeah. They'll be working at Textualize for some months now.

43:05

Alexey

**What did the hiring process look like? Did you just tweet, “Hey, by the way, we got some money. Who wants to work with me?”**

43:13

Will

Well, the first hire was someone that I knew of already. He had an open source project and I exchanged some messages about those open source projects, well before Textualize happened. Of course, I could see all his code and I thought, “Well, this guy is quite good.” Quite by chance, he happened to be in the same city, so I approached him. And quite by chance, again, he happened to be having doubts about the company that he was working for, so that worked out quite well. So he started when the company started – in January. I hired another developer fairly recently, who's a web specialist and has lots of experience. So yeah, I'm quite pleased. The company is very small, obviously.

44:03

Alexey

**Two people, or three people including you, right?**

44:06

Will

That's right. Yeah.

44:08

Alexey

**So you have to do all the HR – pretty much everything, right?**

44:14

Will

Yeah, my wife helps. She's my PA/bookkeeper/HR. [chuckles] Which is a big help. Because those kinds of admin tasks – they take up *way* too much time. So I'm very grateful to my wife for taking those on.

44:32

Alexey

**So it's actually four people.**

44:34

Will

Four people in total, yeah. Three helpers.

# The importance of working on open source for Textualize employees

44:38

Alexey

**When you're hiring people, how important is it for you that they’ve contributed to open source?**

44:49

Will

It's not essential, but it is an indication – what the great thing is when someone contributes to open source is you can see their work. They've got a body of work that you can look through without their knowledge. You can also see the interactions with people. Because we're building public, so the code is just as important as how you respond to PRs and issues. So it gives you a good idea of the employee that you're hiring. So I will say it's not essential – not all developers work with open source, and there's some very talented developers that don't. But it's certainly a check on a list, if a developer has open source experience.

45:40

Alexey

**The two developers that you hired already, do they have this experience? Both of them?**

45:45

Will

The first developer, he did. He’s had quite a lot of experience. The second developer – he had some code, but not a particularly big body of work. But did have lots of experience in a variety of areas.

# The workflow of and contributions to Textualize

46:00

Alexey

**But the way you work is completely open, right? Can you maybe tell us how exactly you organize the work between the three of you?**

46:10

Will

So it's all a standard kind of workflow with GitHub – we're using everything GitHub. To manage the work, we use GitHub Projects, which is kind of like a Kanban Board – you've got the backlog, which has jobs you need to do, and you have a to-do list, which has things that I'd like to be done soon-ish. I can order those by priority and then a developer can pick one that he feels he could do, and he drags out to the ‘doing’ list, works on that, and then we do a peer review. Then, when it goes through the review process, it gets merged and we drag that to ‘done’. It's a super simple system. There's only three of us now. So I think that'll be fine for quite some time. I'm sure at some point, when the team gets bigger, we might have to investigate some other methodologies to manage tasks. But so far that seems to be working for us.

47:13

Alexey

**Do you have occasional contributors? People who use it and say, “Maybe there is a bug,” and they submit a PR?**

47:22

Will

Yes, we do, for both Rich and Textual – because Textual uses Rich. We're refining Rich, making that faster and fixing bugs – also adding features. People can contribute to both projects, Rich in particular. Textual is tricky, because we're really actively working on that in different branches. But we do welcome contributions in the form of code, or issues, or discussions even. [cross-talk] So I think that's one of the strengths of working that way – working in the open.

If you're working with everything closed, you work, work, work, and then you release something which is almost finished, and then you get feedback for the interfaces and the functionality. But if you're working in the open, with open source, you get feedback immediately and ongoing. So by the time you get to release, lots of people have seen it and used it and possibly contributed and found bugs. So I think it's actually a very productive way of working.

48:37

Alexey

**Yeah, I wanted to ask you – if somebody wants to contribute, do you have any good first issues?**

48:45

Will

We don't at the moment. Rich has issues. But Rich is very mature – it's been around for two years. We never seem to run out of issues. So if someone wanted to contribute, that’d probably be the better project to contribute to – just pick an issue. If you think it's something you might fix, we can discuss the approach and then contribute.

Textual, at the moment, is just – we're just working really hard on it. It’s quite hard for someone to contribute at the moment. Not that we would discourage it. But it might be tricky for someone who's new to the project. The plan is, by July, we'll have another release and more stable interface. So that'd be the best point, I think, for developers to contribute.

49:37

Alexey

**Do you have something like GitHub discussions, or Discord, or Gitter, or Slack or something like that?**

49:43

Will

Yeah, we have a GitHub Discussion, so you can jump on that board. We also have a Discord server. I kind of don’t remember the difference – one’s Discord and Discourse. One’s a chat and one’s a forum.

49:56

Alexey

**Yeah, Discourse I think is a forum, and Discord is a chat.**

50:00

Will

Ok, so Discourse is what we have.

50:01

Alexey

**[laughs] Okay. Yeah.**

# Getting your first thousand GitHub Stars (going viral)

50:05

Alexey

**I have a few more questions that I wanted to ask you. They're somewhat related, but not quite. You have a post with the name “Getting your first 1K GitHub Stars”. Maybe you can give us a summary of what is inside this post. How can I get 1,000 GitHub Stars?**

50:27

Will

Yeah. With that post, what I wanted to do was see if I could give some advice for things that work for me. Trouble with these kinds of posts is that something that works for you doesn't necessarily mean that it's going to translate to other people.

50:44

Alexey

**But it might. Right?**

50:45

Will

It might, yeah. The thing is, a lot of people – a lot of developers – will work on something and then I'll put it on GitHub, and then no one will ever visit the GitHub repo. The code can be excellent and it can be very useful. To a certain extent, you have to advertise yourself. If you want to get the feedback, you have to tell people about your project to get them interested in it. I can't remember exactly what I wrote in that post. [chuckles] But I think that that was the nature of it. You have to advertise on various things like Reddit. Reddit is quite good for announcing your project, getting feedback. [cross-talk]

51:26

Alexey

**Is there a specific community, or should you find a community that is relevant to your project?**

51:31

Will

Yeah, there's definitely one. I mean, if it's Python, you go to r/Python [cross-talk]

51:37

Alexey

**I’ve found that with Reddit, when it works, it works pretty well. But usually, I’ve found that moderators are quite annoying. Of course, they want to keep the community safe, but let's say I want to share a piece of my work, but the moderators do not always welcome this work. I can totally understand because I also moderate the Slack we have in DataTalks.club. Did you also have this experience that you wanted to share something, and people get excited, but then moderators come and just remove this thing?**

52:12

Will

That does happen. I think when you push something onto social media, you have to give it something ‘extra’. You can't just post a link and say, “Try this”. You'd have to add some content explaining what this project is, why you’ve built it, give some examples, etc. It has to be more interesting content than just… you don't want it to feel like spam. You want it to feel like something which someone would actively want to read. So if you can do that, you stand a good chance of it getting to r/Python or whatever channel you want. But even then, sometimes I’ve posted things thinking, “Oh, this is fantastic! People will really love this.” And I get like two upvotes and no one comments. I don't know why.

53:10

Alexey

**Two upvotes, two downvotes. Right?**

53:12

Will

Yeah. [chuckles] So you just have to keep pushing your content out there, try to make it more interesting, and then get feedback that way. Eventually, if it's an interesting project that other people would want to try, you will get a bite, and you will start to get some traction, and you will get GitHub stars, and responses to your posts.

53:36

Alexey

**The funny thing with GitHub stars, you mentioned that the project may have excellent code, a very good code base, good test coverage, great documentation, but only 10 stars. At the same time, there are projects that don’t do anything, but it has, like I don't know, 100,000 stars. I think the message in that project was the best code is the code you never wrote – something like this. It's intended as a joke project, I guess. [chuckles]**

**It's funny how much attention this project received, while there are projects where people put a lot of effort into. So that project may be something where the person just did it in half an hour, put it out there and then I guess the tweet went viral or something, or maybe a Hacker News post or whatever. But then when you constantly put out of good work and nobody notices, it can be discouraging, right?**

54:39

Will

It can be. It can be very discouraging, yeah. You have to be a little bit persistent and also recognize that every project has a certain number of people that might use it, so you have to work within that. The thing about Rich was that it was a niche, but quite a big niche. The niche was Python developers who use the terminal and want it to be prettier. It's quite a broad niche. That's why it attracted lots of stars. If you have something that's very specific, a bit of code which might be of interest to maybe a dozen developers around the world, you're not going to get many stars. But if you get 10 stars… [cross-talk]

55:21

Alexey

**You’ll be lucky to get 10 out of those 12. [chuckles]**

55:25

Will

Yeah, exactly. So you've got to normalize it to be a percentage of your bubble. So if you get 10 of the 12 developers to start using it, that’s excellent.

55:38

Alexey

**And I guess Twitter helps, because you constantly share what you’re working on. Then maybe one of these 10 tweets you share, for whatever reason, (I don't know how Twitter works) Twitter decides to show it to more than 10 people – it can be like 100,000. Sometimes you can make two sort of the same tweets, but in one case, it has only like 500 impressions and in the other case, it has 50k impressions. Right?**

56:10

Will

Yeah, it has to reach a critical threshold. You have to have, maybe, someone who's got – I don't know how it works, exactly – I'm guessing that someone with a lot of followers retweets it, and then Twitter sees that as a higher value thing. And then that gets retweeted. And then it goes viral, reaches a critical threshold, and then starts to spread just like a virus. Just like COVID, but not as unpleasant. [chuckles]

56:40

Alexey

**[laughs] Okay, so I guess if you do this everyday, then the chances that one of these tweets might get noticed, even if the rest do not. Like in your case, you did not put this tweet out with the intention of getting noticed by VCs, right? You just wanted to document your journey.**

57:00

Will

Document the journey and get feedback from other developers, even if it's to say “good job” – feels good. They might give you a suggestion, which you feed back into what you're working on. So the process makes what you're working on better – it motivates you and gives you ideas.

# Suggestions for those who wish to start in the open source space

57:20

Alexey

**If somebody wants to start their open source career, or start working on open source, is there something you would suggest that they do?**

57:30

Will

Probably find something which solves a problem that they have. The nature of problems is that it's not just one individual which has it. If you're solving a problem, you'll find lots of other people with that same problem and they might appreciate your work. So yeah, find a problem that you have, solve it, and then see if other people also have that problem.

57:57

Alexey

**Would you suggest open sourcing everything we work on? Just put it on GitHub? If it's possible, of course – if it's not something you do at work. Of course, don't open source that unless your employer agrees. But in your free time – would you open source everything you work on in your free time?**

58:15

Will

I think the only reason not to open source something is if you want to profit from it in a way that you can't share. If you're building a website or service or something that you'd plan on charging for – if you make that completely open source, you've given away the thing that can make you an income. In an ideal world, I'd love everything to be open source. But practically, some things will always be proprietary – at least for a while – at least for the point where it can make the developers an income. But once you reach that point, you could make it open source then. So I would say open source as much as you can, even if it's just some code you just tinkered with – you can put it on GitHub and it might be of interest to someone else. And it might snowball – it might become a bigger project.

59:15

Alexey

**And you can maybe start a company.**

59:18

Will

Maybe, yeah. I had no intention of starting a company when I started working on Rich. I just wanted to print “Hello world” in bright magenta. [chuckles] That was how it started.

# Finding Will online

59:31

Alexey

**[chuckles] Okay. I think that's all we have time for today. So I want to thank you for joining us today, for answering all the questions, for sharing your experience. Maybe before we wrap up, before we finish, if people have questions what’s the best way to reach out to you?**

59:51

Will

Probably Twitter.

59:54

Alexey

**So we'll have the links there. Okay, I guess that's it. Anything you want to say before we end the call?**

60:03

Will

Nothing comes to mind. It’s been fun.

60:07

Alexey

**Okay. Thank you, Will. It was definitely fun. And thanks, everyone, for joining us. See you soon! We have an event tomorrow as well, so check it out.**

60:19

Will

Click the link and subscribe.

60:20

Alexey

**Yeah, of course. [laughs] And also like. Likes are important. Comment. What else do you do? What do I need to say? Oh, and tweet, of course.**