EPISODE 986

[INTRODUCTION]

[00:00:00] JM: Software has changed the way that the world functions, and the rapid pace of change has made a difficult to know how to navigate the new world. Knowledge workers who want to keep advancing in their careers develop a strategy of continuous learning in order to adapt to these changes. O'Reilly Media has existed for almost 40 years providing resources for the technical consumer.

As O'Reilly has expended the product line from books, to conferences, to online learning, the business has grown slowly, but steadily. That business trajectory stands in contrast to many of the software companies that are finally structured to either grow rapidly or die. Today, O'Reilly has a large impact on the software ecosystem. Software professionals congregate at O'Reilly conferences. Enterprises pay O'Reilly to educate their employees, and O'Reilly continuous to grow into new product lines, recently acquiring the interactive learning platform Katacoda, which can be used to learn about Kubernetes and other popular technologies.

In a previous episode, we discussed Tim O'Reilly's book *What's the Future.* In today's show, Tim returns to the show to discuss his experience building O'Reilly and how business philosophy contrast with much of these assumed wisdom of software company building.

We have partnered with SafeGraph for the SafeGraph data hackathon challenge. We're giving away $4,000 in cash prizes as well as SEDaily and SafeGraph swag. SafeGraph is a geospatial data company which curates a dataset of more than 6 million points of interest. I've been doing shows SafeGraph for several years and it's a company that intrigues me. SafeGraph has a high-volume of location data and you can build apps and data science projects with that data. I've you've been looking for a creative opportunity to explore large datasets, specifically with the potential to win $4,000 in cash prizes, this is a great opportunity. The hackathon is hosted on FindCollabs, and to enter, go to findcollabs.com and sign up. If you're planning your own hackathon, you can check out FindCollabs Hackathons. Whether you're running an internal hackathon for your company or you're running an open hackathon so that users can try out your product, FindCollabs Hackathons is a tool for people to build projects and collaborate with each other. It’s a company that I started to allow people to find collaborators for their software projects and our hackathon products allows you organize your hackathon participants to make your hackathon as productive as possible. You can check it out at findcollabs.com.

[SPONSOR MESSAGE]

[00:02:45] JM: Redis is a fast in-memory database system. Engineers have been using Redis for more than a decade because of its reliable object caching, but that's not the only use case of Redis. Redis can be used as your operational data store for queuing, streaming and other data applications.

We recently had an episode of Software Engineering Daily with Alvin Richards of Redis Labs in which Alvin described the use cases of Redis, and I enjoyed learning about the flexible architecture and how Redis uses memory and persistence to create an API that solves a variety of problems. You can listen to that episode or you can go to redislabs.com/sedaily to find out about how Redis can help as a data layer for your microservices.

Redis Labs is the company that makes Redis Enterprise, which offers performance, reliability and professional assistance with your Redis instances. If Redis is on the critical path of your application, go to redislabs.com/sedaily and learn about Redis Labs as well as some of the design patterns for Redis that you might not have seen before. That's redislabs.com/sedaily.

Thank you to Redis Labs for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[00:04:09] JM: Tim O’Reilly, welcome back to Software Engineering Daily.

[00:04:12] TO: Glad to be here.

[00:04:12] JM: The last time we spoke, you were thinking about how to capture the opportunity of upscaling, and this was in the context of upscaling both low-skilled workers and high-skilled workers, people who are already in technology, for example. There are many ways to be upscaled. There are books, and videos, and mentorship, and conferences. Is there a single best way? Is there an ideal curriculum through which an individual can be upscaled?

[00:04:41] TO: I don't really think about it necessarily through the lens of curriculum. In a lot of ways what we really have to start to do is to think about what technology makes possible in learning. How do we learn things today or how do we gain skills? How does technology upscale us?

I was invited recently to keynote at a conference called EDCRUNCH in Moscow. I spent some time thinking about how I would introduce this story, and I started with this wonderful New York Times Magazine article about the knowledge of the streets and monuments of London, which is generally considered one of the most difficult exams in the world. People study through years. Literally, you must become a human GPS in order to pass the exam. The exam is a half-day where they throw you you're at this corner and you have to go to this place. Recite the turn by turn directions that you would use to get there at this time of day. People literally spend years on motorcycles, on foot, on bicycles walking the streets of London memorizing every turn.

I asked this audience. I said, "If you wanted to accelerate the process of people learning to get around in London, is the answer better training for how to master the knowledge?" They answered no. Today we actually have a device that we outsource that knowledge to, and I used the example I just arrived in Moscow for the first time. I have very quickly able to orient myself, get anywhere. The critical point is that even though I originally use the GPS to get around pretty quickly, I could find my way back to all these places that I wanted to go.

It is a kind of – It's actually something. I was originally a classist in learning Latin or Greek. You use something that's called a trot. A trot is actually a parallel translation of this classical text. In fact, the lobe classic sound of Oxford University is one side it's Greek, on the other pages, the facing pages, the English, and you can basically refer back and go, "Okay. How does all this work?" Eventually you kind of learn to do without that aid.

A lot of people think that the GPS trains people to not learn that way around, and you can do that. You can ignore things or become less skilled. I think if you use it correctly, it can be a great aid for helping you up-skilled. The question I post, yeah, using that analogy of Google Maps. What is the analogous technology for software developers?

[00:07:40] JM: The state-of-the-art in what a software engineer has had access to to have that onboarding or education tool or guidance tool that coincidentally also gives you education. Overtime, that's gone from a book, to perhaps a website, to maybe a conference. There are a variety of tools.

The state-of-the-art today you could argue is a guided learning process that to some extent synthesizes the previous pieces of technology we have and to some extent is completely new. For example, there is an online learning platform that you acquired recently called Katacoda. Can you explain the value of Katacoda and how it compares to this GPS analogy that you've just drawn?

[00:08:33] TO: Well, let me actually back up a little bit because I don't actually think that Katacoda per se is analogous to GPS. For software engineering, what's analogous to Google Maps is most likely GitHub. If you think about it, when you say, "I want to do this thing," which is the equivalent of I want to get from A to point B. The critical knowledge is actually encoded by somebody into a software library perhaps, or some reusable piece of code. That's the equivalent of just follow the directions. It's like I want to do this mathematical function. Load NumPy. You don't actually have to know the algorithm anymore. You just have to know which function to call.

As I said, I spent some time thinking about this analogy. So then I said, "Well, what role do we play at O'Reilly?" Because what we used to do is a little bit like teaching people the knowledge. As you said, a book, a conference, a video. Pursuing this analogy, I started thinking a little bit about comment that was made to me by, of all people, an incarcerated person at San Quentin. He was in a coding program called the Last Mile. He was about to get out and he said, "I want to do a startup." Basically the Last Mile teaches incarcerated people to code and I came in to give a talk to them. He said, "I used to work in Fisherman's Wharf, and these people, they can go anywhere, and these people who can take them anywhere they want to go, but they don't know where to go."

In a lot of ways, that's the analogy to what we do at O'Reilly today, is much more helping people understand – Again, actually, there're a lot of complexities in this analogy and I'm still trying to think it through. But let me just give you a great example. Many times the question isn't how do I get to a particular place, is I want to get to a kind of place.

Again, using this GPS analogy, I just came back from a holiday at a wonderful resort in the south coast of Mexico called Playa Viva and I would never have known to say how do I get to Playa Viva? What my wife did was she said we want to have – She actually posted it on Facebook. She said, "I want to know – We're trying to put together an event for about 30 people. We're looking for a small place that we can take over."

She kind of put out some general ideas and did this social search and somebody pointed over this thing. Then we go to Trip Advisor, and Trip Advisor, we looked at this thing. It's got something like 4,500 reviews and 4,400 of them are 5-star reviews. I've never seen anything like it. We go, "Oh! Well, that's clearly our place." I reached out to the guy and we ended up having a marvelous time. But there's a lot of things where you go, "I want to do something like this."

This big range of ways that you have to think about this problem of I want to do this thing. How do I do it? What we're trying to build at O'Reilly is a range of answers to that question, because sometimes you know exactly what you want to do, but you don't know to do it. In which case, we're really doing a lot more work on the search function in our online platform so that we can just get to answers, for example.

You say, "I want to do this task." Again, we've done this in books before. We have the cookbooks. The Pro Cookbook, of the Python Cookbook, the PHP Cookbook, whatever, where it's like, "I want to do this task. How do I do it?" This one or two-page quick explanation. We have our hacks books, which were slightly bigger, more complicated tasks.

Way back in 1992, I wrote a book called *Unix Power Tools* that was, "How do I do this thing?" and there's the small answers. Now we're algorithmically starting to surface that stuff so you can ask these specific questions. Anyways, the things that I'm thinking about are, "Okay. This set of questions which range from the very general where you may want advice." So, "Hey, wait. We have a case study in our online platform where somebody says, “I want to introduce Agile into government. Has anybody else ever done this?” You go, “Yeah.” We can point them to somebody from the United States Digital Service or somewhere else or saying, “Hey! Yeah, we did Agile in government.” That’s a high-level request. A little bit like I want to find a resort that’s kind of like this.”

Then other times it’s all the way down to how do I write this particular regular expression or how would I do gradient decent, or what’s a hyper parameter in AI and how do I set them? These very specific questions and these very general questions. I think what we’ve tried to do which is I think very different from many other online learning platforms is we haven’t basically said our fundamental job is to take you on this sort of guided tour of some – We’re going to take you from A to C, or D, or E, or F. We’re going to basically do some of that. We do have courses. We do have live online trainings that they say, “Hey, you want to do this thing? Here is the step-by-step stuff. Work along with me.” But we also have this playground that is a lot more like GPS or the web where you’re basically able to ask general questions, get answers and then dig deeper. It’s really much more self-directed. We’re trying to put our users in the position of the user of Google Maps rather than saying, “We’re going to teach you to be a London black cabbie.”

[00:14:39] JM: If I understand you correct, you’re saying that there already exists an embarrassment of riches when it comes to ways to learn to do something, to learn to do something particular.

[00:14:53] TO: But not just ways to learn to do something. Ways to do something. I think that’s an important distinction, because in many ways, you learn by doing, and that’s something that we’ve always been very much at the heart of who we are as a company. Back in the early days we were a documentation consulting company, then we were a book publisher, then we were a conference company, an online company. But all of what we were trying to do was say, “Hey, there’re people who know how to do something, and you want to do it too. We’re going to show you how to do it.”

The sort of a slightly different emphasis when you come from documenting a task than it is from when you’re saying, “Well, you don’t know anything about this, so we’re going to teach you to be a software developer,” for example.

[00:15:47] JM: Are you saying that the vision is now you want to help people to understand how to explore the landscape and how to consider new things?

[00:15:57] TO: Absolutely. That's always been part of it. But if you think back to the early days of O'Reilly, there were few people – There's new technology and very few people knew how to do it. We would learn it and we'd write it down and other people were kind of following along. It's still a lot that way. But we built this mental model of what online learning is about, which I think has really been shaped too much by things like coding boot camps or these people who don't know how to code and we're going to teach them from ground zero. Whereas an awful lot of the cutting edge of our industry is people who already know enough to get – They know how to drive. They just don't know how to get to a particular place that they want to go to.

You asked about Katacoda. The reason why we bought Katacoda is because it really fits squarely into this vision, because one of the things that is unique about Katacoda is it's not just – Yes, you can build these learning paths and learning environments where you're able to say, "Hey, you want to do this thing? We're going to teach you about how to do this thing with Kubernetes. So go through this scenario."

First of all, that idea that Katacoda has of what they call scenarios is very much like what we did in our cookbooks years ago. In fact, when we rolled out Katacoda in the platform we actually converted a bunch of material from some of our cookbooks and just put it into Katacoda.

I'll explain what Katacoda is. It's very similar in many ways to Jupyter Notebooks, but it has a broader scope of things that you can do and the environment is more integrated. Basically, it's effectively a description of a process with embedded code that you can execute.

Here's the thing that got me super excited about it. I think about myself as – I don't code on even a monthly basis anymore. I mean, I code occasionally, and you forget things. I think about when I got married in 2015, I built a wedding website. I immediately went, "Well, how do I do that?" I go, "Okay. Let me look. There's a bunch on GitHub," right? I downloaded them, "Oh! They're using Jenkins." I kind of have to figure out a bunch of things about I never used Jenkins before. But it's pretty straightforward and there's some documentation in the site and it's sort of code that I can look at and I can modify. I go, “Oh, yeah. I just take out their image and put in my image and take out their words and put in my words and do a bunch of stuff.” But there were a bunch of things that I went back a month later and I wanted to change something and I had forgotten it all and I had to go back and look at the documentation again.

When Katacoda came along I thought about, “Wow! There’s this whole world of occasional tasks, for example, when you think about our corporate customers and our online platform used a lot by corporate customers. You think about something like – Somebody is saying, “Well, we use this particular set of libraries.” They’re onboarding new people. “We use this particular coding environment. Here’re our policies and procedures.”

Now you imagine building that in an environment where you can actually execute it, because that integration of documentation and executable code is super interesting. I’ve actually been interested in this idea for decades, literally. I think that one of the things that’s most interesting in software is this, in some sense, the moving and shifting of the boundaries between what the computer does and what the human does.

Back, I guess it was 1998, okay. That’s 21 years ago. I wrote this – I started writing. Actually, it was 1998 and then up to about 2003. Actually, the piece I’m thinking I wrote in 2003 was called *Hardware, Software, and Infoware,* and it was really around this idea that in the early stage of computing, the human machine interface was very close to the metal. There were guys back in the early days who literally programmed by flipping switches on the front of a machine, right? It’s like, literally, they were setting bits.

Then when I started, my very first manual I ever wrote was an assembly language programming manually. It was like move this data into this register. Then you have more and more high-level languages. Then you get all the way up, and the time I wrote this piece, I really spent a bunch of time in the late 90s and early [inaudible 00:20:41] thinking about why scripting language were starting to takeoff? These things with late binding; Perl, Python, some others that are no longer really around that much, like Tcl.

It was this great line from this guy, Hassan Schroeder, who is Sun’s first webmaster where he said, “Perl is the duct tape of the internet.” I thought about that, this sort of occasional programming. It’s like you go to a conference and they use duct tape to hold the wires down because, guess what? They’re not going to be tomorrow. It was this idea that the interface had become much more dynamic. There was this other aspect about it that I thought was really interesting, which was I started thinking about HTML as a real breakthrough. I did this contrast between, say, Microsoft Word which was sort of the state-of-the-art PC software at the time and this new thing, the web, which was really starting to take off.

I said, “Look. In an application like Microsoft Word, you have little bits of human speech embedded in software code. This menu.” There are prompts, and basically somebody has sort of put in the human speech into a program. Now look at how HTML reverses the paradigm. You actually – With CGI, which was the original way that you had dynamic content in a webpage, what we’re doing is we’re actually embedding programs and programmed actions into a human document. Webpage, you click on that link and it fires up a program. We've inverted the paradigm.

I think if you look at the way that we’re continuing to evolve software, we are increasingly bringing it closer and closer to where people are effectively going to be programmed. Again, they’re not even programming. They’re basically – They are calling stored procedures. If you can see the continuity between a programmer saying, “Load this library and then execute this function.” Really, we’re basically at a higher and higher level, loading more and more code. Then you kind of think about the end-user who says, “Okay, Google.” The point is I think we will get to a point where we have the speech interfaces that really work pretty well. In fact, they already do work pretty darn well for a lot of things. These things that you used to have to enter commands, and now you can just talk.

I guess what I'm saying is the guiding idea that we’re playing with O’Reilly or one of the guiding ideas is this idea of the continuum of the human machine interface and the ability to help people to understand how to use the computer to help them with a task or to accomplish a task, all along that continuum. I think a lot of the guiding metaphor that many people use when they think about online training is a little bit like we’re going to accelerate the London black cabbie getting the knowledge. What we’re trying to do is saying, “No. No. We’re going to build better and better interfaces where people can actually do things and have to say less, do less, explain less, learn less in order to get the computer to jump through hoops.” The reason why you want to do that is the more you can make easy, the more you can reserve your actual year learning and hard brainpower for things that are now hard.

I always loved one of the Perl sayings back in the 90s that I was loved was the aim of Perl is to make easy things easy and hard things possible. That’s a lot of what we try to do. We try to make easy things easy and hard things possible.

Out of that far cutting edge, you really may have to learn radically new skills. When people get a book like Aurélien Géron's wonderful book on Tensorflow and scikit-learn, he's got this incredible pedagogy for explaining how you wrap your head around deep learning, because, hey, this is new hard stuff. But there's a lot of things where you just kind of go, “No. Actually, I don’t actually need to understand if you’re a particular type of programmer. You may just be – All you need to know is which function do I call.

[SPONSOR MESSAGE]

[00:25:22] JM: Today's episode is sponsored by Datadog, a cloud scale monitoring service that provides comprehensive visibility into cloud, hybrid and multi-cloud environments with over 250 integrations. Datadog unifies your metrics, your logs and your distributed request traces in one platform so that you can investigate and troubleshoot issues across every layer of your stack.

Use Datadog’s rich customizable dashboards and algorithmic alerts to ensure redundancy across multi-cloud deployments and monitor cloud migrations in real-time. Start a free trial today and Datadog will send you a t-shirt. You can visit softwareengineeringdaily.com/datadog for more details. That's softwareengineeringdaily.com/Datadog, and you will get a free t-shirt for trying out Datadog.

Thanks to Datadog for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:26:22] JM: One thing I like about talking to you is most of what you talk about is this high-level abstract magic of computing, the almost religious aspect of what is so cool about computing. But together with that collection of words that comes out of your mouth that is very high-level and abstract, you have a history of making incredibly smart tactical decisions in how you have built the business. I think there are people who make very good tactical decisions in building their businesses but may not have being a big, high-level abstract view of the majestic nature of computing and there are people who have very high-level views of the majestic nature of computing who may not be so good at making tactical decisions. You seem to have done a good job of finding a marriage there. In particular in the nuanced financial and just tactical business discipline you've taken to your business, we talked about this a lot in the last episode where just said that O'Reilly is not a financialized company in the sense that your products included in that set of products is not a stock ticker symbol. It is not a price of a stock.

With that in mind, I'd like to know a little bit more about how you think about the mechanics and the selection of an acquisition like this, because the vision you just laid out, it's very beautiful. It's also very broad. There're a lot of companies you could choose from in selecting something to acquire that might fit as a puzzle piece snugly into the portfolio of O'Reilly.

Can you just give me a little bit more light on the lower-level aspects of why you made this particular acquisition?

[00:28:05] TO: Well, first off, there aren’t a lot of companies that a lot of technologies that do what Katacoda does. This Jupyter Notebooks and this Katacoda in our opinion, and we support both in the platform. We’ve been fascinated with Jupyter for years. We actually ran a Jupyter conference for a couple of years. When we encounter Katacoda, what we found so fascinating about it was just how easy it is to integrate Jupyter. It’s still little. It’s like if you look at the Jupyter integration or our platform, there are Jupyter notebooks associated with some of the books where the code is all executable, but you go into this different environment and Katacoda has got a lot in common with this thing that I've been in love with since the early days of the web, which is this sort of easy back-and-forth between code and explanation of the code. It's just I think it's an environment that – Also, in addition, it makes it easy to do things like shell programming to do a lot of the things that are in that sort of system administration, network administration, security area and not just data science and things through data heavy where Jupyter excels, although you can do that as well.

It just really increased our interactive capabilities, and I think the thing that we're doing that's – We are doing a lot that isn't just the learning space, but I think we're really also seeing it as a way of exploring how do we turn the platform into something that just helps people do their work, because that's always been our goal, to be kind of a knowledge utility. You use Google every day. You use GitHub every day. How we use O'Reilly every day? We think that there're a lot of things that we do where you go, "Okay. I'm going to set aside some time for learning a new skill."

Actually, if we look at the usage patterns that we already have of the way people consume our content on the platform, a lot of it is like they're looking for an answer to a problem. They got their coding environment in one window and they've got O'Reilly in another window because they're looking for answers. Where are they going to type on the next line? They're not looking. They're not sitting there going, "I'm trying to learn how to program." They're saying, "I'm actually trying to build this machine learning model and I'm not quite sure how to set that next parameter, or I'm not quite sure how to optimize this or what does my data pipeline look like?" They're trying to get very specific answers.

The other thing I guess that we do, we obviously have this environment where we built a marketplace. I mean, that's the other thing that's really critical about the O'Reilly online platform. We don't produce all the content. We're always looking for core capabilities that we can open up to our marketplace partners because that allows us to get useful content in areas that we don't really do. We've never been a certification company, but Pearson is one of our partners. Guess what? They do all the certifications. Can we give them better tooling to build better content that's more useful for our users?

We did that when we introduced live online training and we said, "Okay. We put together this package of tools. Obviously, we actually have live synchronous training. We opened it up to partners on the platform so we would have training in areas that we ourselves didn't want to do. Now as we're looking at these tools for more interactive, not just say interactive learning, but interactive doing, supported doing. Basically, we wanted to have it be part of the capabilities of the platform so that we can say to our partners, "This is something you can use to build on."

As I said, our roots as a documentation, we're really on some ways coming full circle, because that was the thing that I think distinguished O'Reilly books from the very beginning, was that we were a documentation company, whereas other people were like, "Hey! We're writing a book about –" Again, people writing books about software programs and they were sort of documenting it or this is how Microsoft Excel works or whatever. We were like, "You want to get on the Internet? Here's how you do it?" It's like it was focused on the thing you want to do. It was always sort of doing first. "You want to put up a webpage? Here's how you do it. Oh, wait. You have to –"

Again. I think back on when we did that first book on the Internet, the whole Internet user's guide and catalog in 1992, and then everybody was doing books on the Internet. What do we do? We had a book, "Well, how do you set up a webserver? How do you set up your DNS?" It was soon driven by the fact that, "Well, if you want to have a website, well, here are the things that you have to know about. You have to be able to set up a database. You have to be able set up –" Driven by a sense of what people were trying to accomplish.

[00:33:06] JM: Not to change the subject too dramatically, but you started the company I believe 37 years ago at this point.

[00:33:13] TO: Depends how you count. I started my – IN partnership with another guy, we started documentation company. It would've been in 1978. Then the company – We were in partnership for five years. We broke up the partnership in '83, and that's when O'Reilly Media proper was founded. You can either say –

[00:33:34] JM: 42 years.

[00:33:35] TO: Yeah.

[00:33:36] JM: Something like that. Anyway, when you started O'Reilly – Or how long did it take you to start to have a vision for how the company would grow into other markets, or did you always have that from the beginning? When did it shift from being kind of the mere idea of a documentation company to something where you saw the opportunity for something bigger?

[00:33:59] TO: I guess it was the first piece I wrote that would – It was both my first piece of advocacy out there on – It was called Documentation and the Future of UNIX. It would have been 1984, so 35 years ago I guess. It was basically something that I think I've always been good at, which was just a piece of pattern recognition, which was that I recognized that something was changing about the software industry and that UNIX, eventually Linux and the Internet, were changing the rules.

Before that point – Again, this is ancient history for many of your listeners. Every company made their own computer with its own operating system. The PC was a bit of a revolution and that there was a standard architecture in a single dominant. When I started, that wasn't even really happening. It was like it was Digital Equipment Corporation and Prime Computer and HP.

What I saw with the early adoption of UNIX was like, "Wow! All these people, they still have different hardware." I was working for a company called MassComp, and Prime, and Data General, whatever, and they were all starting to have, "Well, we're going to compete on the basis of our hardware, but we're all going to use UNIX, or least it will be an option on our platforms." I said, "Oh!"

One of the things that I recognized there was, well, if it was only one version of the software, then there would be this sort of idea for reusable documentation. I was sort of making the case that, "Well, we could just take –" First of all, it was a lot of these stuff at UNIX had come out of a research environment. So there really wasn't much documentation. I'm still thinking of early books like [inaudible 00:35:43]. The only documentation on [inaudible 00:35:45] was a research paper out of Bell Labs. It was like six or seven pages where they were kind of saying, "Hey, we wrote this cool program," but it really wasn't anything you would – Then it was a man page, whatever. Wow! We can really improve on that.

When X window system came out in 1988, it was like, "Wow! This is a demonstration project that they want people to build on. Documentation that's pretty lousy. We can make it better." What I recognized was that all of these companies were in parallel trying to do the same thing, which was we have to improve this crappy documentation for you this free software we're adapting. We basically said, "Well, you could do it yourself or you could just get it from us."

We actually had – It was a breakthrough I think when it was sort of there were some version of the X window system. I don’t remember which one, but it was the next consortium meeting. MIT had started something called the MITX Consortium and Bob Scheifler was the director. At some meeting somebody said, “Well, what are we going to do about documentation?” Scheifler said, “Don't worry about it. O'Reilly will do something.”

We literally put all these documentation departments out of business, because we had realized that once all the companies were using the same software, we could produce commercial books that would fill the void. The thing that’s sort of interesting, if you look at the long arc of the last 30 years, there was this sort of golden age for commodity software where there were a lot of publishers producing books and documentation and the vendors didn't have to do that much.

Of course, the way that the publishing market evolved consolidation and various other things, that became less true and companies have ended up having to do much more their own documentation again. Part of what we've understood is – In particular, as people are working with more and more fragmented environments, the industry has really changed. We first saw this around JavaScript frameworks, because there were so many of them. It was just this jungle of possibilities and you might be at one company and you’re using Jango or you’re at another company and you’re using React. The challenges of having sort of a service single path through that jungle became much greater. Every company was having to roll their own again.

Again, I think our marketplace approach has really helped with that because it makes it possible to have much more documentation. We allow – Our corporate customers can even upload their own documentation in another platform, these various ways that we have tried to deal with that. But I think that it’s still a long way to go.

Anyway, the point in response to your question is I think the thing that we've always been pretty good at is we look at these big patterns in the history of the computer industry and we asked ourselves, we try to sit back and we try to say, “Where is this going?” I wrote about this in my book *WTF*. How do you do pattern recognition on the present in order to predict the future?

It’s not really that you're predicting the future. It’s that you’re identifying vectors that are adding up to take you somewhere and then you’re looking for signs that one vector or another is being reinforced so that when I wrote that first piece on documentation in The Future of UNIX, it was the beginning of noticing something and then noticing that Linux was taking off and free software of all kinds was taking off. You start seeing these vector start to add up and you start going, “Okay. We’re going to pile on that.”

Right now we’re in this sort of transformation that’s around machine learning becoming everyday technology and how that changes the software development lifecycle, a softer development process. Peter Norvig gave a great talk about this at our first AI Conference, which whenever people ask me, “Well, what should I look at?” We have that talk in our platform and it’s like, “Hey, how does this change programming?” Companies that are trying to adapt machine learning, they have to develop a whole new set of skills again, and those are skills that are really organizational skills. It’s not just at the level of, “Well, you need to have your developers learn about Tensorflow or PyTorch.” It's also that you actually have to understand where is our data going to be. Who collects it? Who curates it? How do we clean it? What kind of people do we need to build a data pipeline into these processes?

[00:40:17] JM: That stuff that's more easily digested in conference format in many cases, I think. The organizational touchy-feely side of things.

[00:40:26] TO: It is. But again, you have to have the big picture in order to know how do you give people context? There’s this great quote from this guy, Edmund Schlossberg, that I’ve always loved and tried to live by. He said, “The skill of writing is to provide a context in which other people can think.”

So often, people go astray because they don't they don't take the time to get the big picture. That’s one of the things that we've always tried to do in O'Reilly books, which is to have an introduction. My working theory when I first so developed a lot of our book formats was that there’s really two kinds of knowledge in those books, and one is this big picture. How does this thing work? How do all the pieces fit together? Then there’s very specific knowledge, which is, “Well, what function do I call or what command do I issue?” and that you wanted to give people enough contexts pretty quickly because many of our users are in a hurry that they can then drop in anywhere later and they would know enough to be – They’d be oriented.

It’s kind of like a little bit like when you go to a new city. How do you get a big picture of this is the area where – These great restaurants. This is where the best shopping is. This is where – Oh my God! This museum is amazing, or whatever it might be that you’re interested in. This is how you get out and that’s a great park. You have that level of orientation and then you kind of like, “Well, how do I get there? What do I do when I get there? What are the opening hours?” There are all these analogies there that you can start to say, “Well, how do those this play out in the way people learn and do things online?”

[00:42:10] JM: Your company started as a media company. I mean –

[00:42:15] TO: Not really. We started as the documentation company. We renamed the company O'Reilly Media. It was originally O'Reilly and Associates. We were literally a band of consultants. We named it O'Reilly Media.

[00:42:26] JM: Documentation is a kind of – Whatever.

[00:42:28] TO: When people think of media, they often think of advertising. We've always been – Yeah.

[00:42:32] JM: Okay. I just meant in a sense that eventually you evolved into a technology company where you had to build technology.

[00:42:37] TO: Yeah, that’s right. That’s correct. Yeah.

[00:42:39] JM: Tell me about crossing the chasm from being a publisher of content to that plus more of a platform.

[00:42:49] TO: That's been a big challenge for us, and I think that we were held back for a number of years by the fact that it was a joint venture. Our online platform, we started it in 2001 and we invited our biggest competitor, at the time, Pearson Technology Group in as a partner. It was set up as sort of a separate company, and we didn't take full control of it back till 2014. We really should've done it much sooner, because any joint venture, you’re always sort of –

[00:43:17] JM: Wait. What was it originally called?

[00:43:19] TO: It was called Safari Books Online.

[00:43:20] JM: Safari Books Online. Okay. That was with you and Pearson.

[00:43:23] TO: Yeah.

[00:43:24] JM: Oh gosh! That must've been really hard to unwind.

[00:43:26] TO: Yeah, it was. If we could have done it a few years earlier, it would've been much better, but it just meant that it was harder to be hands-on with the platform because we had these people who were reporting in the both companies. It was a separate company.

Once we took control of it again, we kind of had to really get in there and start seeing what was going on. We went through a couple of false starts, but then I think we really started to hit our stride. The thing that – This is the other side of what I am spending a lot of time on apart from things like really trying to understand how we build around the future of learning. But the other is what does it mean to be a platform?

If you look at a lot of things that I've been writing publicly, like the piece that I've written in courts. They're really about what's wrong with the idea that seems to be the Silicon Valley playbook today, which is winner takes all. Because if you really want to have a platform, it's got to be good for the people who use it.

There was this great line which Ben Thompson who writes this great news article called Stratechery calls the Bill Gates line. He has a wonderful post. It's a conversation between Bill Gates and Chamath Palihapitiya, who was the head of so-called Facebook platform. Bill probably said to him, "That's not a platform. A platform is when the people who use it a get more out of it than you do." It's funny to have Bill be the one saying that, but wisdom may come — of course, Microsoft, that was true originally. But then Microsoft started to take too much out of platform and that led the lots of problems. But that is in fact the key idea.

I've been writing about the way that it looks to me by looking at the Google's financials and so on that they're taking more and more the value out of the web and it's becoming less vibrant as a platform. They've been – There was this sweet spot where they were really allocating value very fairly to lots and lots of people and now they're allocating more and more the value to themselves just like Microsoft did on the PC, and Apple's doing the same thing in the App Store.

I've really been working very hard to understand how we don't do that at O'Reilly. It's very easy to give yourself privileged access. Very easy to say, "Oh wow! We just introduced this feature. Wow! We're taking more of the pie. This is great." There are different kinds of – Again, this different terminology. Ben would probably disagree with me on – Maybe I'm thinking about – I may be mixing up whether – Anyways, maybe somebody else. But anyway, this idea of platform –

[00:46:06] JM: Okay. Ben Thompson is standing for every tech company.

[00:46:08] TO: Yeah. Right. Platforms versus aggregators and so on, but there are people who are pure enablers. You think about somebody like Stripe. They don't control anything. They've just provided some infrastructure library, or Twilio, or whatever. Whereas someone like Google, or Facebook, or Amazon, whether you call them an aggregator or a platform, they're really in the position to decide who gets what and why, which is they control the economy of their platform. It's real temptation to cheat. Microsoft did it on the PC. They gave themselves privileged APIs. They also where they were saying, "Well, great! Wow! These great applications. We'll do those too." Then they privileged their own applications, and now you see the same thing with Google, these various kinds of content that really are commodities where you kind of go, "Okay. Everybody's building on – Whether data that's provided by the U.S. government. Well, Google or stock data, or whatever. This is public data. That's fair.

But then there are things like, say, Trip Advisor, which spent 15 years sort of building up this database of customer comments and then Google says, "Well, we've got to do that. We're going to actually – We think we don't want to pay these guys. This way we'll just use our platform position to put our content first and our reviews first." I kind of think that's very much the Microsoft play.

Back to O'Reilly, we do have a privileged position because we're both the platform for our online learning platform and we are a participant. Still, it makes me smile all the time and it must have been 2-1/2 years ago now. Whenever we introduce our live online training, Laura Baldwin, our president who really runs the company day-to-day calls an emergency meeting of our exec team. She says, "We have a real problem. We introduced live online training. It's been so successful that Pearson's revenue is going to fall by half in the next month." This is a problem. It wasn't like, "Wow! We just took – We introduced this new feature." First of all, when we introduced it, we socialized it through all our – Through our partners.

[00:48:13] JM: Sorry. This was when Pearson was still part of it?

[00:48:16] TO: No. Pearson it still is a part of it. They're not an owner now.

[00:48:20] JM: Oh, okay.

[00:48:19] TO: But they're still on the platform as hundreds of other publishers of content. We really are a platform, but we introduced this new feature, but we went all-in on it. We introduced 100 live online trainings and Pearson introduced 10. It was such a successful new feature that the revenue is allocated among publishers in proportion to usage and we were just – It's actually usage times price is a somewhat complicated algorithm for it. But the problem was we sucked a huge amount of value out of the platform.

What we end up doing was radically reducing the price of our online trainings until the other people on the platform could catch up, because our goal wasn't to like take the value. It was to keep them producing value. I've been thinking a lot about this idea of two-sided markets, and maybe some of – Actually, I started thinking about this a lot with Uber and Lyft and some of these online matching marketplaces where you go to track drivers, you go to track passengers. That was the first piece I wrote for courts. It was a critique on split-scaling. Because there was so much capital they were able to say, "Oh! We're just going to optimize for the passenger side and we're going to screw the drivers."

They treated the drivers as sort of a fungible resource. In fact, if Capital had been less scarce, we might've got a different balance of who gets what and why, which is that fundamental question of economics. You go, “Okay. Well, we’re going to have to pay this much so the drivers are going to have to pay this much, or the passengers.” In a way, I think we’ve forgotten that a platform thrives when suppliers thrive.

At O'Reilly, we’re spending a lot of time. When we introduced a new feature like Katacoda, how do we get other people on board? How can they use it? How do we teach them to create content in areas that we’re not going to do it? I think most of the platforms I see out there, maybe they start by bootstrapping some content from outside providers and then they’re like, “Well, actually, this is a really lucrative area. We should do this ourselves.” I think our fundamental commitment to this as a platform play I think is something very distinctive.

[SPONSOR MESSAGE]

[00:50:43] JM: Over the last few months, I've started hearing about Retool. Every business needs internal tools, but if we’re being honest, I don't know of many engineers who really enjoy building internal tools. It can be hard to get engineering resources to build back-office applications and it’s definitely hard to get engineers excited about maintaining those back-office applications. Companies like a Doordash, and Brex, and Amazon use Retool to build custom internal tools faster.

The idea is that internal tools mostly look the same. They're made out of tables, and dropdowns, and buttons, and text inputs. Retool gives you a drag-and-drop interface so engineers can build these internal UIs in hours, not days, and they can spend more time building features that customers will see. Retool connects to any database and API. For example, if you are pulling data from Postgres, you just write a SQL query. You drag a table on to the canvas.

If you want to try out Retool, you can go to retool.com/sedaily. That's R-E-T-O-O-L.com/sedaily, and you can even host Retool on-premise if you want to keep it ultra-secure. I've heard a lot of good things about Retool from engineers who I respect. So check it out at retool.com/sedaily.

[INTERVIEW CONTINUED]

[00:52:19] JM: A question that's somewhat tangential to that. Coming back to the idea of your nonfinancialized company, it is hard to compete for top talent if you are not one of these financialized companies, because a lot of the best talent can go to a place like Airbnb or Stripe and get not only great cash positions but can get lottery tickets that can mean so much money in a financialized company.

As you've shifted to becoming more and more of a technology platform, in some ways your growth is going to be bottlenecked by how many good strong engineers you can get, or I guess the strategic leadership. How do you reconcile that competition for talent with the financialized companies?

[00:53:11] TO: Well, I think that I won't deny that it's a serious challenge. We're competing with companies that effectively have a currency that is –

[00:53:19] JM: Super money.

[00:53:20] TO: Super money. Yes. What I've called super money. First off, idealism goes a long way. There are people who are sick and tired of working for companies that they don't feel good about. But also I think there's also a bit of fatigue at the lottery nature of this thing. Yes, there are companies. There are definitely companies where you can get a guaranteed huge return. Amazon will pay you in stock and it's likely they keep paying off. On the other hand, if you're kind going to a startup, that's becoming less and less – Because the big guys are in fact doing it wrong exactly in the way that I'm saying. They're kind of taking too much of the value. Think about some of the companies they have invested in where it was sort of like, "Oh, yeah. Amazon says that's a really nice idea. Sell to us, or we'll just do it." I think a lot of people have felt like they've been screwed over by the venture capital model, in which they're basically cannon fodder and chips and betting in a casino mindset.

That's part of it. The other thing that I think really helps us we have a distributed work environment so we can hire people who are not here in Silicon Valley where everything is so expensive, and there are good engineers all over and we work a lot remotely and we've kind of developed it. I think it's the combination of values, the combination of being a distributed company.

I mean, there's no question that would we like to be able to get people more easily? Absolutely. I also say, my wife, Jen Pahlka, runs a nonprofit code for America, where trying get people where there's absolutely no prospect of it, an exit or outcome, and that you still find good people.

When she was at the White House and recruiting people for the United States Digital Service, these people said, "Well, I can make Larry and Sergey a little bit richer or I can make it possible for people to get healthcare." When you think about the healthcare sort of rescue team that came out of Google where they're like, "Oh, yeah," and then they thought it was just going to be a temporary thing and then it was some of like, "Whoa! We are really impacting people's lives. At Google, we're not doing that anymore." I think people want to have a positive impact and people want to make a difference.

[00:55:43] JM: Yeah. Since we're kind of near to the end of our time, I want go into some further flung question, because you do write a lot of editorials, or not a lot, but you read enough editorials to make me curious. I follow your Twitter, the concept of publisher versus platform. We've explored this a little bit, but more in the context of O'Reilly, which is pretty tame relative to the questions faced by some of the other publisher/platform? People have vastly divergent views on how sympathetic they are towards Twitter and Facebook for the kinds of content moderation and censorship issues that they have.

Do you have any concise perspective for what these companies should be doing differently? Do we even have enough experience with this domain to legislate from the outside or do we kind of have to just give credence to Facebook and YouTube and Twitter and just assume they're doing their best? What can we actually say confidently about what they should or should not be doing?

[00:56:52] TO: Well, I guess I would say a couple of things there. First off, the way we talk about the issue is really badly broken. This is not about free speech ever. It's not about free speech, because if you think about how social media works, anyone can speak. But the question is not about censorship. It's not about stopping somebody from saying something. It's about how much do you choose to amplify what they have said?

Chris Cox who was the chief product officer of Facebook I thought put it pretty well. He said, "I think our issue is anybody can speak, but we don't have a responsibility to give them –" Anybody can get on their soapbox on Facebook, but we are not responsible for gathering their audience, and we shouldn't be.

The way I think about Facebook, and Twitter, and Google to the extent they do this, YouTube, they are absolutely responsible for their curation. They're absolutely – Just like if New York Times chooses to put fake news on its front page, you go, "Hey! You just curated that stuff up to the front," and yet that is what Facebook is doing all the time, is curating stuff. Yes, their individual front pages, but they're curating stuff to the front, and they should be held accountable for that. That being said, I think they are working very hard on it.

One of the really good examples of this recently, there was a new study came out and somebody said, "Oh, everybody talks about how YouTube radicalizes people. Well, we just did this new scientific study and we found that, in fact, YouTube is actually making people more moderate when they –"

I reached out to Rene [inaudible 00:58:46] who'd been one of the early people who was raising the flag on this and she said, "Yeah, because they've changed what they do in response to all of the red flags that were going out." They have made progress.

[00:58:58] JM: Hold on. Do you ever, where are these people getting the data that they're talking about for doing these gigantic studies? It seems so questionable to me that you could actually get enough data to really say this with confidence.

[00:59:09] TO: No. There's a very good book by Yochai Benkler and team at Harvard called *Network Propaganda* where they were just using. You can public Twitter data, for example. They were studying – That was another thing, and I just –

[00:59:23] JM: Do you know what percentage is bots. You don't know what the denominator is.

[00:59:27] TO: I mean, the point is, weathers it's bots or humans, you can kind of look at what happens to it on the network. It's like it does it matter if it's a bot that got on your front page or if it was from a human. That's not what they're trying to measure. They're just trying to say, "How does it propagate?"

[00:59:43] JM: No. I'm saying the bot. If they're measuring whether a bot got radicalized, like if they're saying, "Oh! Before radicalization, this was after radicalization," whether or not it's a bot matters.

[00:59:54] TO: Well, sure. I mean, I think, I'm not sure you would even consider that a bot would be radicalized because the bot has a purpose.

[01:00:02] JM: No, but that's the point, is how would you identify the dataset for a study like this? If you're trying to identify how humans – Anyway.

[01:00:07] TO: Yeah. I am not deep into that particular set –

[01:00:09] JM: Okay. Different question. Sorry. Not for you.

[01:00:11] TO: I do think that we are holding these guys feet to the fire is important, but recognizing how much progress they have made and how much effort they've put in towards it is important. Because if you contrast them with mainstream traditional media, there are companies, Fox is a notable example, that actually has a business model around actually canonizing this information back to Yochai's book, *Network Propaganda*. They literally looked at the spread of fake news and they said, "Look. It actually starts equally on the left and the right." On the left, mainstream media tends to tamp down rumors. They start to spread. Then at some point mainstream media doesn't kind of raise them to the top. On the right, it gets up to the top and Fox treat it as real news, and that's the big difference.

Here's a company that whose business model is to spread misinformation. I guess that's always been true. You think back at William Randolph Hearst and the Spanish-American War, you want a war, I'll give your war. But we are in an era of yellow journalism. I think that there is a mainstream media conspiracy to blame tech.

[01:01:32] JM: I agree.

[01:01:33] TO: And that we need to basically not buy that, because the tech companies are doing more than any of the mainstream media companies to actually fight this. Mainstream media companies are still – They're like, "Hey, man." They're eager for their clicks. They'll do clickbait headlines even the best — now, that being said, there is this issue that this fundamental Achilles' heel of the social media engagement model that it tends to reward sensationalism, but there is a problem that companies like Facebook are trying to moderate content that they are also incentivizing, and that's a serious problem.

[01:02:21] JM: Completely agreed. Love your summarization there. There was the case of Cloudflare making a decision to shut down a white supremacist site that was hosted on Cloudflare servers. Do you think hosting companies should be forced to be treated as content moderation platforms? Should they be total open blank platforms? How should they answer this question?

Again, this goes back to I think a lot of people don't really understand the concept of censorship. Censorship is something that a government can do. Censorship is something that someone empowered us. If somebody says, "Look. I don't want your business because I don't appreciate your values," I think that's a perfectly reasonable consumer decision. It's no different than somebody saying that – Again, it's sort of tough, because there are cases where the government intervenes. For example, these are protected classes. You can't discriminate on the basis of race. You can't discriminate on the basis of sexual orientation. You can't discriminate – Again, that's the government, which is something that expresses our collective will making that determination.

That's why Mark was saying, "Hey, we would really like the government to give us more guidance on some of these things." But in the absence of that guidance, there's no reason why Facebook, for example – My personal advice to Mark was, "Look. You got to stop pretending that this is some kind of vast democracy where your users get to decide in this free speech." It's like, "You're a king. Your job is to be a good king."

It didn't go anywhere, but I think if he had values around like, "No, there're a bunch of stuff that I really don't believe." It was like, "Hey, if Rupert Murdoch can say, "I don't believe in climate change and I'm going to basically not cover any of it and I am going to–" Nobody talks about that as like, "Oh my God! How is it that Fox is censoring any news about climate change and amplifying climate denial and they're not having their feet held to the fire? That's their – Because everybody says, "Well, this. They're right as a media company."

Why would it be any different if Facebook said, "No. Actually, we're taking a stand on some issues." I think that they're a little bit lily-livered hiding behind the free speech idea, and I love it when somebody says, "No. We're not going to host a white supremacist site. We're not going to host hateful content. We are going to actually take action." I think we need more of that, not less of it.

Of course, if there are – Where you get into issues is when companies are in a position of great power, when somebody has a monopoly. But I actually think many of the – There's a lot of this a lot of mao-maoing of social media companies by the right-wing saying, "You are censoring us." When in fact the evidence is the opposite, they're actually – In fact, I just read Steve Levy's forthcoming book about Facebook, and there's a lot of evidence from inside Facebook that they were basically like in the 2016 elections. They thought Hillary was going to win. So they've spent a lot of time trying to appease the right-wing. Why piss them off? It's not going to matter. Actually, it's a matter of opinion, but there were opinions of people on the inside who thought, "Yeah, we are basically appeasing the right wing." It wasn't – Whereas the right-wing is saying, "Oh, you're censoring us." No. They're actually amplifying that.

I think that the idea, we will get better at all this. I do think that is true. But I do think it is – Because it has become a subject of kind of cyber warfare, that's going to be hard. We have a long way to go before it gets better. In fact, it may never get better.

I don't know if you've read the Neal Stephenson's fall, but it opens with this so great kind of like little section where they're really talking about people who this is massive disinformation campaign on Facebook and people just don't want to know the truth. Not on Facebook. On the equivalent of social media.

[01:06:33] JM: That sounds like reality.

[01:06:34] TO: Yeah. But literally, where this is sort of fake nuclear explosion, people won't go to the town to see that it’s actually still there. It’s just the willing desire of certain priorities to believe certain things and have consensual – Sort of like, “Anyway, whatever.” That is something that is problematic. I do have to say, if you have the perspective of history, you do realize these things can get so seriously out of control and you do end up with the fall of a civilization.

There’s a wonderful book called *The Swerve*, which is really about the rediscovery of science in the Renaissance, but it starts with kind of like this beautiful picture of the devotional literacy and knowledge in the Roman Empire and this sort of turning point where a female Egyptian mathematician, Hypatia, stoned to death for being smart. That’s kind of in some ways the telling of Greenblatt's book, that is the turning point that leads to the dark ages. Where basically there was a turning away from knowledge.

I think we have big parts of our society look at something like climate change where people are saying, “We don't want to believe the scientists. We want to believe –” And there are people who have their political or commercial interests pushing that narrative and there are people who swallow it and then we see this sort of turning against expertise. There’s a lot of people playing with fire and this huge issues at stake, not just the politics, but look at the environment and you look at where the future goes. Helping people to navigate this new landscape is really important.

[01:08:27] JM: Tim, O’Reilly, thanks for coming back on.

[01:08:30] TO: Thank you.

[END OF INTERVIEW]

[01:08:39] JM: Today’s show is brought to you by Heroku, which has been my most frequently used cloud provider since I started as a software engineer. Heroku allows me to build and deploy my apps quickly without friction. Heroku’s focus has always been on the developer experience, and working with data on the platform brings that same great experience. Heroku knows that you need fast access to data and insights so you can bring the most compelling and relevant apps to market.

Heroku's fully managed Postgres, Redis and Kafka data services help you get started faster and be more productive. Whether you're working with Postgres, or Apache Kafka, or Redis, and that means you can focus on building data-driven apps, not data infrastructure.

Visit softwareengineeringdaily.com/herokudata to learn about Heroku's managed data services. We build our own site, softwaredaily.com on Heroku, and as we scale, we will eventually need access to data services. I'm looking forward to taking advantage of Heroku's managed data services because I'm confident that they will be as easy to use as Heroku's core deployment and application management systems.

Visit softwareengineeringdaily.com/herokudata to find out more, and thanks to Heroku for being a sponsor of Software Engineering Daily.

[END]