EPISODE 598

[INTRODUCTION]

[0:00:00.3] JM: Ten years ago, a biology researcher would have been limited by the software tools that were available. Most of the electronic record-keeping was done using Excel and other general purpose tools. Benchling is a suite of software tools that were designed to simplify the lives of life science researchers. Benchling helps with sample tracking, experiment design and workflow management.

Sherwin Yu is a engineering manager at Benchling. He joins the show to discuss the workflows of the life scientist, how experiments are designed and managed using software. Life science researchers in both academia and industry use Benchling, and Sherwin spends time talking to both of those groups and understanding what they need from their software tools. We also talked about the impact of CRISPR, robotic cloud laboratories and other future developments.

I also want to announce that we're looking for writers for Software Engineering Daily. We want to bring in new voices. We're focused on high-quality content about software that will stand the test of time. You are listening to content about software engineering right now, you probably also read content about software engineering. Do you want to write? Go to softwareengineeringdaily.com/write to find out more. We're looking for part-time and full-time software journalists. We want to explain technical concepts and tell the untold stories of the software world. We're looking for writers who only want to produce a single piece of content and we're also looking for people that want to produce a series or an in-depth investigative piece.

We just launched a new design at softwareengineeringdaily.com, so if you'd like to work with us go to softwareengineeringdaily.com/write. You can also send me an e-mail, jeff@softwareengineeringdaily.com.

[SPONSOR MESSAGE]

[0:01:54.4] JM: Every team has its own software and every team has specific questions about that internal software. Stack Overflow for Teams is a private secure home for your teams'

questions and answers. No more digging through stale wiki's and lost e-mails. Give your team back the time it needs to build better products.

Your engineering team already knows and loves Stack Overflow. They don't need another tool that they won't use. Get everything that 50 million people already love about Stack Overflow in a private secure environment with Stack Overflow for Teams. Try it today with your first 14 days free. Go to s.tk/daily.

Stack Overflow for Teams gives your team the answers they need to be productive, with the same interface that Stack Overflow users are familiar with. Go to s.tk/daily to try it today with your first 14 days free. Thank You Stack Overflow for Teams.

[INTERVIEW]

[0:03:07.6] JM: Sherwin, you are an engineering manager at Benchling. Welcome to Software Engineering Daily.

[0:03:12.0] SY: Hello. Thanks for having me, Jeffrey.

[0:03:14.0] JM: Benchling is a set of tools for doing life science research, biology research. When the company got started, the research tools for life scientists were quite outdated. When I was in college, I worked in a biology lab for a while. I remember using lab notebooks and spreadsheets. If I were to try to do that work now, I would just be driven insane, because it would feel like a very outdated workflow. Talk about some of the outdated workflows in the life science sector that Benchling is trying to improve.

[0:03:53.1] SY: It's funny that you mentioned that, because I also had quite a bit of lab experience in high school and college and felt similar frustrations with the tools that people were using. I think that was one of the big motivations for our founders as well, who both had a lot of extensive lab experience in college.

To give you a picture of what lab tools and lab workflows look like nowadays, a lot of it is still done in paper and pencil. You will literally be required in lab depending on the lab that you're

working in, you might actually be required to just make all of your recordings in paper and pen. That is not good for a number of obvious reasons, like everything else has moved to being done on the computer, but this stuff has not.

For instance, if you spill coffee on your lab notebook, that might just be entire years' worth of research gone. Or if someone steals it, that's entire years of research gone. On top of that, you basically also have issues like it's hard to search your lab notebook. If you actually want to have a sense of like, "Oh, I think I've done an experiment on a similar topic before, but it might have been four or five months ago. How am I going to find that?" You have to literally just flip through your lab notebook until you find that date.

Then beyond that, there's also just like it's hard to have full context of what is actually happening, because if you're struggling to keep up the pace of all the notes that you're producing over time, you're just not going to document as much. If you had some tool that would help you track experimental conditions, or the results that you're seeing and you don't have to fill that all in by hand, you would just spend a lot less time logging and writing up your experiments and more time actually doing the research and thinking about the problems. That's an example on the lab notebook side.

I would also say another good example would be the actual tools that people are using. In biology, people, scientists will be working with DNA sequences, right? As you probably know, DNA sequences are just these long strings of bases. The letters ACTG and extremely complicated permutations. There are some people who still basically use effectively a glorified text editor to look at these bases.

It's like they will open a gene, which might be thousands or hundreds of thousands of bases and then try to find, "Oh, I want something from the 5,000 position," and then they will just copy and paste that by hand. It's like, "Oh, I want to select bases from 5,000 to 6,431. Then they will copy that and then maybe send that to a colleague an e-mail, or maybe they'll paste it into a spreadsheet to do some subsequent analysis, but it's a mess.

Basically, you would probably want some dedicated tools to actually help you analyze, edit, filter and understand your biological data as well. Then I would say one last thing is just collaboration

in the lab is quite challenging. To give you an example, actually when I was in college we would run our experiments so you're doing these gel electrophoresis experiments, where you basically run DNA on a gel and then it fluoresces under UV light, so you would bring it under this fancy machine and then it would – you would basically take a image. You would take a scan of the gel.

This is all in one integrated lab machine, which is probably running Windows XP or something. It's not connected to the internet, so you then have to plug in a USB drive. It's exporting the image in some super outdated image file format, so then you have to drag that onto your USB drive, bring that back on over to your laptop and the entire process takes 10 minutes. Wouldn't it be great if all of these super expensive machines would just understand how the internet works and how modern collaboration software works?

[0:07:58.6] JM: You gave four examples there. The first one was simply taking the spreadsheets and the lab notebooks and making them into domain-specific software. This is a very low-hanging fruit. This is the classic, you look at any industry and find the spreadsheets that some knowledge worker in that industry is using and figure out the macros and the columns and the rows that they're using, and then build a CRM or some similar software around that. That's one of the things Benchling does.

The second thing you mentioned was the usability of the data, so you have these super long strings of DNA and you have people opening up text editors finding the 5,000th position, copy-pasting from the 5,000th to the 5,050th position and pasting that somewhere else. That's obviously an error-prone type of action to take. You mentioned the problem of connectivity between different lab modules, that gel electrophoresis.

This is a hardware, specific hardware that's dedicated to pouring agar gels into and then doing protein measurements of the weights of different the breakdowns of molecules, I think in protein that run through that gel. Then end result is some image. You have all these – that's just an example of a piece of hardware in the lab that is not smart.

You have these collection of problems. Basically, you could walk into a lab and find that those problems and find probably 50 other problems with the software stack in the lab and the integration between the software and the hardware. Why has laboratory software been so

immature and so mediocre for so long? Why haven't these problems been addressed prior to Benchling was started in what? 2010 or something?

[0:09:58.9] SY: 2012. It's been about six years, yeah.

[0:10:02.2] JM: Why did it take so long? We've had life science labs with computers in them for a really long time.

[0:10:07.0] SY: To be fair, when I say it's like the DNA editing example, I mean, it's not quite that bad, like it is. It's not just you're opening it in notepad. There are going to be some helpful features, like the existing tools that were around before Benchling, but still a pretty error-prone experience. I think the actual copying of I want to find the specific range, there's actually a pretty common occurrence if you're browsing a sequence online, because it's just a – it's a pretty outdated webpage and it'll just like, "Oh, here's the full contents of this DNA sequence for this gene." It will just spit it out at you, right?

There have been approaches to all of these things in the past, I would say. I think the frustration is that even the existing – so it's not people have just been twiddling the thumbs doing everything manually for the past decade and a half, I think the concern is that the tools are simply not very good, they're not very usable. Even if a existing tool is trying to address the problem of like, "Oh, how do I design and edit a DNA sequence?"

The existing tools are pretty unpolished. They're not designed – the way I would explain it is it looks like 90s software, like 90s Enterprise Java applet software, if that makes sense. You're asking for why that's the case. I've thought a lot about this. I think one big reason is that at the end of the day, biologists and scientists are biologists and scientists. They aren't engineers, so they don't have the mindset of, "Oh, I have this problem on my workflow. I have the attitude, or the capability, or the skills to actually go on and just change it or improve it."

Whereas, that's a very, very different mindset from what we have in engineering and tech and especially in Silicon Valley, and in startup culture where it's – just think about all of the startups that are basically making tools for other startups, or other tech organizations to use. Or I think it's a very common engineering mindset to say, "Hey, I have this broken tool. How can I make it

better? How can I make it faster?" I think it's simply a matter of the fact that we as engineers have the capacity to improve our tools and tooling is a really, really important part of engineer's workflow.

[0:12:21.8]JM: Yeah. I mean, it's second nature to us as software engineers to think about how to improve the workflow. It's not a second nature – I mean, if I walk through a building, I'm not sophisticated enough to know well why are the steps in that place? Why is the elevator in this location? Why are there only two elevators as opposed to four elevators? I would not be well-equipped to improve a building that I walked into. I'm sure if an architect, or a civil engineer, or an architectural engineer were to walk into a building, they would see all kinds of flaws with that situation.

I mean, when I was briefly in biology before I was in computer science, I didn't even consider that, "Dear God, why is the workflow like this?" Because I didn't even have the mentality of the software engineer to improve those things. Benchling has developed the set of tools for improving the workflow of a life scientist. What does Benchling do to change the experimentation process? Give me a picture of the before and after. Maybe you have an example experiment in mind that could be useful, but help me understand how Benchling changes the experimentation process.

[0:13:38.8] SY: Sure. I would say Benchling has a pretty profound effect, both for individual scientists and for teams of scientists working together, and then more and more as some more and more recent investments in our product is focusing on actually entire R&D organizations, so teams of teams of teams working together.

I can start at the individual scientist level. What you basically have there I would say is – I mentioned common things for a biologist to be doing or they say, "Oh, I need to design the DNA sequence, or I need to inspect some DNA, and then I also want to record my experiments in a lab notebook." We have this molecular biology suite, which is a DNA editor, a protein editor and built-in analysis tools, and then also this electronic lab notebook.

Each of those get you as a scientist. If you are trying to design, or edit DNA, you can think of a DNA editor as just – it's like what a IDE is to your engineering to your source code, right? It's

like, there's syntax highlighting. In DNA land that would be, “Oh, we want to highlight areas of interest on this DNA sequence.” This promoter is regulating this gene and then the gene will be highlighted and the promoter will be highlighted.

Or we want to say this entire area code is for this protein and that'll be an annotation on the sequence. A lot of this is using just existing portable DNA sequence files. We support a number of those files and it's just like – so you can always get your data in and out of Benchling, because there are existing DNA and sequence and protein sequence repositories.

Other nice things that the DNA editor provides are just access to really, really common and some of the more advanced analysis tools as well. I would actually say one of the cool things about working at Benchling is that frequently, we get to implement a lot of cutting-edge bioinformatics algorithms. If you've heard of the along the buzz around CRISPR, which is a pretty revolutionary way to allow gene editing, it's like if you actually want to design a CRISPR experiment, it's a pretty computationally expensive operation.

Right now, I guess, I would say maybe five years ago when CRISPR experiments were just taking off, people – there was a lot of interest in it and people would try to say, go to like the professors who are working on CRISPER, they will – when they publish the paper on CRISPR, they'll also host a website and it's like a PHP script on their academic domain. Of course, that's not a super great experience.

What we end up doing is we'll take some of these cutting-edge algorithms that are – it's all the all the stuff is fully described in their papers that are publishing. Then we'll basically productionize them. We'll add a really, really nice usability layer on top of it, ensure that it runs and scales in production, give it high availability, things like that. When you're using the DNA editor, or the protein editor on Benchling, you have direct access to all of these things.

Whereas before, what you would frequently have to do is like, “Hey, I'm working with this DNA sequence. I want to run a sequence alignment, which is basically I want to see how similar these two or three or four or any number of sequences are to one another. I want to align them together.”

In cases like those, you would have to pre-Benchling, you would have to open an existing utility that does sequence alignment really, really well. You would have to paste in your sequences and then you would run it. Or you have to go to a website, in some professor's academic homepage and then paste in your DNA sequences and run it. Because Benchling is trying to be this all-inclusive research platform, we can just run it directly for you, because we have already implemented those algorithms and you just – Benchling is aware of all DNA sequences that you have on Benchling and you can just – in a type ahead say like, "Oh, I want to align these four sequences," and you can just type them by name and then it'll auto-populate.

[SPONSOR MESSAGE]

[0:17:46.4] JM: Azure Container Service simplifies the deployment, management and operations of Kubernetes. Eliminate the complicated planning and deployment of fully orchestrated containerized applications with Kubernetes.

You can quickly provision clusters to be up and running in no time, while simplifying your monitoring and cluster management through auto upgrades and a built-in operations console. Avoid being locked-in to any one vendor or resource. You can continue to work with the tools that you already know, so just helm and move applications to any Kubernetes deployment.

Integrate with your choice of container registry, including Azure container registry. Also, quickly and efficiently scale to maximize your resource utilization without having to take your applications offline. Isolate your application from infrastructure failures and transparently scale the underlying infrastructure to meet growing demands, all while increasing the security, reliability and availability of critical business workloads with Azure.

To learn more about Azure Container Service and other Azure services, as well as receive a free e-book by Brendan Burns, go to aka.ms/sedaily. Brendan Burns is the creator of Kubernetes and his e-book is about some of the distributed systems design lessons that he has learned building Kubernetes.

That e-book is available at aka.ms/sedaily.

[INTERVIEW CONTINUED]

[0:19:22.2] JM: You mentioned that designing a CRISPR experiment being computationally expensive. What can you tell me about the process of designing a crisper experiment?

[0:19:31.5] SY: When you're designing a CRISPR experiment, you basically – to give some background CRISPR, it's basically a way to make very, very precise DNA edits, so cuts and insertions and the way it works is that you have a guide, which is a short DNA sequence, so maybe 20 base pairs, or 20 letters, and you attach that guide to essentially these protein scissors that will allow you to edit DNA specifically.

The guide is what – it's like a homing this – so you're attaching this homing component to this DNA editing component and the two things in it as a system will basically find the specific place in your entire genome that you want to edit and then make the edit that you want. Now this is where the computationally interesting portion happens. If you design a guide that is too long, then it might be overly specific it won't match anything. If it's too short, it might match in too many places.

If you imagine the entire genome of a human is 4 billion base pairs. If that were all just random, then they are going to be a number of – like the one place that you're trying to edit is probably just going to randomly occur some number of times, depending on how long the guide string that you're looking for is. What you end up having to do and there are lots of fancy – the biology is considerably more complex than just this, but you basically need to balance a number of factors in terms of finding how you can actually match your guide against the genome.

Typically a CRISPR algorithm will say like, “Hey, I want to make edits at these particular positions, and I want to design the best possible guide that maximizes the specificity for this region and minimizes what we call the off target score, which is how likely is this guide going to bind in places that I don't actually want to be editing,” because if you make a edit somewhere else, that's really, really bad, because you're cutting up DNA.

That would be an example where computationally it's intensive and we actually have done some pretty interesting performance optimization on this. We use in the back and we use AWS

Lambda to spin up these super massively parallelizable genome searches. There's entire blog post on this, about one of the other engineers at Benchling wrote and I can send that to you. It's actually a really, really fascinating technical problem, because it is – if you think about a super, super parallelizable, but then they're also you have to think about if people want to do these searches in sub-second time, how do you get it down to be that fast across a number of genomes, so some researchers will want to work on the human genome and then we basically want to index the human genome, then some researchers will want to work on a strawberry genome, or etc There are a number of technical challenges and I'll send you the link to the blog post.

[0:22:27.1] JM: Are you developing these gene search algorithms yourself? Or are you able to take these off the shelf from papers that have already been written?

[0:22:35.0] SY: Yeah. We more or less take them off the shelf. We're definitely not in the business of trying to do groundbreaking science ourselves. At the end of the day, we are trying to build a research platform. Frequently, we will make optimizations to the algorithms though, because a lot of times, the white paper, or the methods that a professor is publishing in academia is not super optimized, and they don't need to be, because they don't need to run at a scale or run it at in a production environment.

[0:23:03.5] JM: CRISPR specifically, how is CRISPR changing the life science research field?

[0:23:10.6] SY: I will share my own opinions on this. I think CRISPR is probably the biggest development in molecular biology since the discovery of the structure of DNA. That would be in the last 100 years is probably up there next to discovering the structure of DNA. Simply put it, because it allows unprecedented specificity and effectiveness of gene editing at a fraction of the cost of the techniques that came before.

Gene-editing is something that we were able to do beforehand, like there were multiple approaches and techniques ranging from things that have some specificity, even to a technique called site-directed mutagenesis, which is like, “Oh, I just want to randomly mutate some positions in my DNA.” Of course, that has no specificity. It's basically completely random.

What CRISPR gets you is that it's – I would say it's like giving you a surgeon's tools with that level of specificity and precision, compared to beforehand where you might be trying to operate on the DNA with a sledgehammer. What this actually allows you to do is, oh, I mean, the implications are profound in disease. There are a lot of just genetic diseases, where you just have this mutation in your DNA. If you have it, then you'll have some inherited disease and at least a lot of human suffering. What CRISPR will allow us to do is go and edit that gene and then basically edit DNA in living organisms.

[0:24:42.9] JM: Then there's complementary technologies, like the gene drive, right? Because then you need to be able to propagate that gene to the body, right? You have to edit it and then you have to propagate it.

[0:24:53.6] SY: Yup, yup. The gene drive would be something like people talk about this for say malaria a lot, where if we just introduce this into the mosquito population and then over time this gene will just propagate throughout the entire mosquito population. I think there are – I'm by no means an expert on that, but it's certainly pretty fascinating. In human therapies for instance though, one area where CRISPR has really shown a lot of promise is addressing diseases in any part of the body where the cells live very, very long, where there's low cell turnover.

Say like nerves, or in the eye where your cells basically are going to stick around for a long time, because then when you make an edit to the DNA within a cell, it's going to stick around for a long time.

[0:25:40.7] JM: Okay, so I get that this is a useful tool and the implications are profound. What's the state of the tooling? How usable is CRISPR today? What can you actually do with it? Are the experiments at the toy level, or are they actually at the therapeutic level, in the deployment level? Give me a feel for the landscape.

[0:26:02.8] SY: There are a number of CRISPR, I would say biotech startups that are in the process of commercializing their therapies. These all basically popped up in the last five years or so. Just to give you a sense of how the pharmaceutical industry operates, typically the time scale for bringing a drug to market is on the order of 10 to 20 years. Those would be more traditional small molecule drugs, but this just more general trend of biologic drugs. Biological

therapies such as things empowered by CRISPR has really, really changed the game in terms of how we actually deliver and do research on these therapies.

I would say there are a number of companies that are actually doing this in clinical trials and they're still ongoing research and actually improving the technique itself, so scientists have still been discovering sibling systems to CRISPR that might offer even more specificity, or have some of the drawbacks addressed, or work differently in different conditions.

[0:27:10.1] JM: Are the people using the Benchling product, just to revisit Benchling is this suite of tools that your day-to-day work is you are a life scientist, you're a biologist and Benchling is at the heart of your workflow. How many of these people are using CRISPR tools, or I mean, can they just explore CRISPR tools? Do they have to actually be doing some experiment with CRISPR to actually make use of the Benchling CRISPR tools, or are there ways to play around with it in the software realm?

[0:27:45.8] SY: Yeah, yeah. That's a great question. We basically do have a standalone CRISPR tool. That means that you don't really need to use Benchling's lab notebook, you don't need to use Benchling's file registry, you don't need to use Benchling's protein editor. It's basically, I have this genome and I want to run a CRISPR experiment. I want to design some guides for it, and you can just do that in Benchling. It's a big of our academic customers actually, because of the buzz of CRISPR, so a lot of people might use – they might hear about CRISPR at a conference, or from a colleague and they might want to just check it out, see how it works.

The first place that they might go to would be academic website, and then they realized it's a pretty bad experience. Then a lot of people end up pointing them towards the Benchling tool. There are a ton of people who are just playing around with CRISPR, and then we also have of course a good number of scientists, both in academia and industry who are using it professionally for the work.

[0:28:43.1] JM: Okay, so let's say I'm in industry, and I'm trying to develop a new therapeutic treatment for, I don't know, name your malady. How am I using Benchling? Can you help me understand the workflow of the enterprise life scientist and how this software suite, the workflow suite of Benchling fits into their day-to-day work?

[0:29:08.0] SY: I mentioned a little bit earlier that we have this lab notebook and this molecular biology suite. I would say those are mostly impacting individuals, right? Now how does that actually scale at an organization? You can think of this as – how about this, I'll walk you through as the complexity of a organization grows, how does Benchling fit their needs, right?

At the individual level, you are probably just like, “Oh, well. I don't really care about collaboration. I just want a nice lab notebook, and I want a nice DNA editor.” We have that. A lot of our academics basically use that, because it's like a grad student working alone, or maybe it's even a student in class and they just want nice tools.

Now if you are say four or five people in a academic lab working maybe under a professor, or under a postdoc, you will have some amount of collaboration. Our tools already support that, so you can have a shared folder, you can see each other's comments inside of a lab notebook entry and you can leave comments for one another. No more e-mailing files back and forth to your team, etc.

Now if you're in a more, like a more established lab, so this might be say like 50 people working together either in industry or in academia, this is one communication starts getting interesting, because now not everyone knows everyone, not everyone is working on the same experiment, and you are going to start seeing duplicated effort, or hey, someone recently did an experiment on this gene and has some result. Where can I find that result?

This is where you basically want to start building a bio-registry, like a database that is a system and record of all of the biological information and learning of your organization. When you get to something like that, you can basically think of it as like a library where you can register individual biological entities. A typical workflow like this might be, I am a scientist, I'm doing some experiment, I'm trying to design new sequence, or design new plasmid. I do that and then I characterize it, get some results. All of that would happen in my private workspace, or it may be in a small private folder shared with some collaborators.

Then after I have really polished that up, then I want to basically share that with the rest of my organization, so then I would register it and put it into our entire lab’s bio-registry. One way to

think about this might be like, “Oh, I am working on a branch and it's going to be pretty rough.” In tech, I would be working in a branch. I can get comments and feedback on the pull request and then eventually we clean it up and then we want to bring it into master. We want to merge it.

This is a way where we say people can have their own tiny workspaces and then once they have fully finished investigating something, they'll share it with the rest of the team. Then this addresses a few problems. One is that now, you actually have a canonical representation of all of the biological data in your lab. If we're all doing research on the same DNA sequence, we should probably have an agreement on exactly which version of that DNA sequence we want to be using as an example.

[0:32:17.9] JM: Is that how drug discovery works, where it's similar to a process of this branching and merging that we see in computer science-based jobs?

[0:32:29.1] SY: Not quite. This would be for a smaller lab, like on the order of 40 people. Drug discovery, like the organizations that have the resources to do something like drug discovery, and this is – this leads me to my next point, is basically like, okay, I'm actually going to have multiple divisions or multiple teams all working on the same research effort, which is I want to bring these promising therapies to market, right?

In that case, this might be even a step up above like a 40-person team, where you actually have teams of teams working together. This is where you would want to basically support. In an organization like this, you will have entire teams dedicated to performing a certain function. For instance, there might be a protein assay team, and their job is basically to receive – to basically support the other research teams by performing certain experiments or analyses on the proteins that are going through the pipeline.

You can think of a larger organization as basically being extremely process-oriented. They have a pipeline. Every team is playing its part, so you need a much better system of tracking progress on number of experiments, or projects across teams and also across the entire organization. The most recent product that we've really been investing in is called workflow management, and it's basically how do you monitor, optimize, organize research and development across a large organization?

[0:33:56.2] JM: The show I did in the past that comes to mind is the one I do with Transcriptic, which was this robotic hardware cloud lab. Essentially, the idea is you have some experiment that you want to run in the cloud and they have all these different machines that are – I don't want to tape together with duct tape and chicken wire, but that's basically what it is. Where you have all these pieces of legacy hardware in the biology lab; you've got PCR machines and centrifuges and gel electrophoresis machines. This stuff doesn't really talk to each other natively.

What they have had to do is to build these interfaces between the different machines in order to stitch together workflows between them and have robotic arms that carry samples between them. I can imagine it would be quite useful to have data from each of these steps plugging into something like Benchline, especially if you are one of these giant companies that's doing drug discovery. At each of these steps, you've got manual data entry. I mean, it's great that you have a piece of software like Benchling that facilitates the workflow, but it still sounds throughout all these steps, you have a human in the loop where the human is recording data and there is potential for inaccurate recordings. What is the state of the integration between the hardware and the software in biotech?

[0:35:31.7] SY: Yeah, yeah. That's a great question. It's certainly a big, big opportunity for Benchling. Also, I think Transcriptic is a super, super cool company. I think there are some pretty interesting parallels between what Transcriptic and Benchling are doing. As far as how we can integrate all of these disparate data sources produced by various machines in the lab, one thing that a lot of our enterprise customers want is basically APIs, right?

Some of the more modern machines basically will provide API access. An example would be like, I want to scan a plate. This is a common operation where I basically have a grid. Say it's a 10 by 10 grid, like a physical grid. Each of those cells is like a container with some sample of DNA, or some biological sample. I could individually pipette volume from each of these things into my machine, and then have it do the reading and then pipe at the next one, have it do the reading, record it, and basically do this manual operation a 100 times.

If this is a large well-funded research organization, why would they pay their extremely expensive scientists a bunch of money to manually pipette liquid back and forth? There are machines that basically do this for you, right? It will basically take a reading of the entire thing at the same time. Then how do you get that data?

This is something that Benchling is currently addressing right now. We're building integrations with these machines that will allow it to talk directly to Benchling. It will just drop those results directly into your lab notebook, and it also associate it with the correct biological entity that is already in your organization's bio-registry. As a whole, I think this is big opportunity both for Benchling and for life science research in general.

[0:37:23.2] JM: Do you have to reverse engineer the protocols of these hardware devices in order to get them to talk to Benchling?

[0:37:31.9] SY: Not too much. A lot of the more recent ones will – it's like provided. There have been cases in the past, in the early days when we we're doing some reverse engineering. I remember one time, this is when we were very, very young, just like I think four people or so. There was a university lab that was at the time, if we could get them to sign on to Benchling, that would have been great. They were like, "Oh, all of our DNA sequences are in this proprietary format. How can we get them into Benchling?" Then we reverse-engineer the file format.

[0:38:01.8] JM: You're an engineering manager at Benchling. How much domain expertise do you need to have in biology to do the job effectively, since you're effectively engineering tools for the biologists?

[0:38:13.4] SY: Yeah, that's a good question. I actually studied both computer science and biochemistry, so I have a fair amount of just a biochemistry or molecular biology background. It's funny, because at least in college, the course that I was least excited about was like wet lab biology, where you actually have to go in on a Wednesday or Friday afternoon for four hours and basically move tiny amounts of liquid.

[0:38:37.9] JM: Painful.

[0:38:38.9] SY: Right. I remember talking to my advisor at the time and I said like, “Hey, I'm mostly interested in the computational stuff, that's why I'm double majoring. I'm never going to do lab work. Why do I have to do all these?” Then they said, “Just take the class.” Nowadays, that's the class that gives me the most that is actually the most relevant for my work, because it gives me a lot of empathy for our customers who are just scientists who actually are doing this wet lab biology.

To answer your question, most of the engineers at Benchling do not have a biology, or biochemistry background, and that's totally fine, because we have PhDs and scientists on staff to help ramp people up, to give them the proper context on biological workflows and such. At Benchling, if you're say working on the marketing, or the customer success team, then it's pretty expected that you have some biological experience.

I think for us, we also have a big culture of learning. Even though a lot of people are trained only as engineers and not us biologists, we do journal clubs where we'll have one of our senior scientists actually just walk through the science behind one of our customers workflows, and people definitely really enjoy learning about the biology and getting exposure to it.

[SPONSOR MESSAGE]

[0:40:01.0] JM: Users have come to expect real-time. They crave alerts that their payment is received. They crave little cars zooming around on the map. They crave locking their doors at home when they’re not at home. There is no need to reinvent the wheel when it comes to making your app real-time.

PubNub makes it simple, enabling you to build immersive and interactive experiences on the web, on mobile phones, embedded into hardware and any other device connected to the internet. With powerful APIs and a robust global infrastructure, you can stream geo-location data, you can send chat messages, you can turn on sprinklers, or you can rock your baby’s crib when they start crying. PubNub literally powers IoT cribs. 70 SDKs for web, mobile, IoT and more means that you can start streaming data in real-time without a ton of compatibility headaches. No need to build your own SDKs from scratch.

Lastly, PubNub includes a ton of other real-time features beyond real-time messaging, like presence for online or offline detection and access manager to thwart trolls and hackers. Go to pubnub.com/sedaily to get started. They offer a generous Sandbox to you that's free forever until your app takes off, that is. Pubnub.com/sedaily, that's P-U-B-N-U-B.com/sedaily.

Thank you PubNub for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:41:44.2] JM: There have been a number of listeners that have written in and asked for more information on engineering management. More conversations around engineering management. Tell me what you have learned as you've gone from being an engineer, to being an engineering manager.

[0:42:01.3] SY: Yeah. To just give a little brief background, I've been at Benchling for about four years. For the first year and a half or so, I was engineer, and then the last two and a half years, I'd been an engineering manager. It's been a really, really interesting ride. I could speak to, I guess, some general learnings around engineering management, or I could also talk about how we do it at Benchling and how the unique position that Benchling is in. How that actually affects. I'll talk about Benchling in particular.

The interesting things about Benchling is that, well first, it's a enterprise software as a service company, and that has a lot of implications. Then it's also not just enterprise. What's like in a pretty technical domain-specific vertical, so life sciences, right? We are life sciences enterprise software as a service. Then the last couple and I'll add to that is that our team and our company's core competency has always been its product.

We hire very, very talented engineers. All the engineers are also very product-oriented, so they participate in product discussions, they have opinions on design. As a whole, Benchling is I think the usability and the product experience of using Benchling is adhere above a lot of our competitors,

These three things have some pretty interesting implications for how we basically do our work, and then also how I think about the team's growth as a manager. To give you an example, Benchling will never be the company, because this enterprise SaaS, that say like has week over a week, like the user count is growing by 20%. It'll never be like a snapchat, or Twitter, or something like that.

What that means is that you have to think about how to keep people excited and motivated at the company, when the company is growing at a pretty steady, but not exponential blowing up rate. There are plenty of things I can talk about that there. Other interesting implications are that, because we're selling to a pretty – I would say slow-moving risk-averse industry, so life sciences, so pharmaceutical companies, biotech companies, that also changes how we think about product development. For instance, like why would a multi-billion dollar pharmaceutical company trust their intellectual property, like all of the science and the workflows that they are doing on this super tiny startup, right?

That has also been a interesting struggle. Our general strategy for that is basically we go up the chain. We first sell to and get buy-in from really, really small companies that might just be a professor who is commercializing his or her research with the postdocs, that's made of like a 30-person team. Really, really do a good job with them, and then figure out what product we need to build for them, and then go for the next tier up on the ladder.

It might be in a hundred person, or like more established biotech company, but still definitely not like a pharmaceutical giant. For something like them, we would probably – like the needs of a larger organization on Benchling are going to be slightly different from the needs of a 30-person company. Then it's a lot of what was called like product code development. Before we even signed a deal with a customer like that, we spent a lot of time really, really understanding that workflows, understanding what they – actually understanding what their pain points are, what the problems are and how Benchling can provide value to them.

It's tricky, because you don't – because we're not a consulting shop, right? We don't want to build a tailored solution just for one particular customer. We have to hear what they're saying and then also align that with Benchling's overall vision, right? We basically have to assign these co-development contracts, because otherwise these larger companies aren't going to trust that

we're actually going to deliver, because why would they, if we're such on a established company?

[0:46:05.8] JM: Code development means you're contracting for them, but you're building software that is your IP.

[0:46:12.6] SY: Yeah. I would say that co-develop means that they are very, very engaged in the product development process. We take their feedback at every step of the way, from ideation, to actually talking to their scientific leads, to talking to all of the people at their organization, and then cross-referencing that against our own internal product division. For many of these customers, we would actually sign a statement of work that would outline exactly what we want to do for them.

[0:46:41.6] JM: That's more about the product development side of things. I imagine you're interacting with the C-suite, you're interacting with the salespeople, you're interacting with the product developers, and you're talking to them about how you're going to translate that into engineering reality. That's one side of your job.

What about the side of your job where you're bringing that vision into implementation? You're interacting with engineers, you need to understand human psychology, you need to understand how to ratchet certain deadlines for the engineers, you need to know how to orchestrate sprints. This is tactical stuff that is not easy. There's no recipe for this stuff, even though people know about sprints and OKRs and typical engineering management rules of thumb bringing this stuff into reality is far harder than – I mean, as I have found. I've been doing some engineering management the last year or so, and it's hard. It's not straightforward. What lessons have you learned about the process of managing engineers that you work with?

[0:47:46.0] SY: Yeah, yeah. I would say that almost all of the unique interesting engineering management challenges that arise that Benchling also are heavily tied to the fact that we are like enterprise life science company. It's definitely the case that the engineering management at a company like Benchling is going to be very, very different from engineering management at a consumer-facing web app, or a consumer-facing startup.

To give an example of that would be well, because we have enterprise clients, that means we also have deadlines. We have a lot of deadlines. That's a tricky situation, because it's a pro because it means that you have very, very clear focused understanding of what you need to do. You can tell that's delivering real value for the company, because if we finish this sprint and hit this deadline, we are literally unblocking this deal, and that deal is directly contributing to the revenue of our company and our company's bottom line.

The con that you have to address or manage is that well, it means that people don't always get to work on what they want and it means that people will frequently be pressed to do things, to take shortcuts, or to not necessarily address all their technical debt. We have to make very deliberate trade-offs between both product polish and technical polish. If we want to do something correctly, it might take additional seven days, but the deadline is in four days, so how do we reconcile that?

[0:49:11.7] JM: I know we're nearing the end of our time. I wanted to ask you about a question that's totally far-flung from engineering and engineering management and Benchling, which is the reproducibility crisis. There is a reproducibility problem in life sciences. Scientific results will get created once through a series of biological experiments. It's often difficult or nearly impossible to reproduce the same result. Was there anything else you want to add about engineering management, or are you interested in talking about the reproducibility crisis for a bit?

[0:49:44.9] SY: I have a few things I could add about engineering management, I guess. Maybe some actual – like the actual insights, or learnings that I've had, because I've been talking about it. Yeah, so one pretty big realization for me as an engineering manager is just how do we think about growth, right? This is super important, a company like Benchling as I mentioned, because we're not growing super quickly in headcount and I think at a company like say, Snapchat where there are tons of people joining every single day and the entire company, like in the early stages the company just feels it is a rocket ship.

You can headcount growth and company growth solves all problems, because there's so much momentum and people just feel excited. When you don't have – when that's not the case, when you're only hiring an engineer every month, or every couple months and the company is

growing slowly and steadily, do you actually have to be a little bit deliberate about how do you make sure people are motivated? How do you make sure people are still learning and how do you make sure that this is a good place for them in their career?

The way I like to think about engineering management is that you basically want – as a manager, I basically have two customers. One is my team and one is the company. I need to make sure that my team is happy and motivated and is growing and all the things that they are doing are aligned with their ultimate personal goals and personal and professional goals. Then I also have to make sure that my team as a unit is delivering on these commitments to the company.

Engineering management fundamentally is about this trade-off of okay, sometimes I will have to sacrifice a little bit of what might be most important for the company in order to give this, to allow one of my engineers to take some more time on a project that would be really, really good for her growth.

When we actually think about growth, I think one good framework that I really enjoyed is basically thinking in terms of breaking down as like craft mentorship, which is just technical mentorship. How are you thinking about engineering problems? How do you implement things? How do you design them? Your familiarity with the tools; so all of that is the actual craft of engineering. I would say there's career mentorship and that's bigger picture, thinking like what are the major achievements that you want to accomplish in the next year? What are the general areas that you want to grow in? What would your ideal career look like at Benchling and outside of Benchling?

Then I would also say there's a number of life skills mentorship, and that's things like, “Oh, how do I relate to disappointment, or to failure? How do you make sure that you are motivated and excited at work? How do you make sure that you are not working too hard? How do you make sure that you're making the correct decisions?” Or a good example is when you feel down, how do you think about that? How does that affect your ability to be a contributing member at your team? A lot of the obviously like emotional intelligence, emotional skills is a last piece that I think a lot of engineers struggle with and can be mentored and can be practiced.

[0:52:40.4] JM: To close off, I want to talk a little bit about the reproducibility problem in life sciences. Scientific results often get created once through a series of experiments, and it's difficult or in some sad cases, completely impossible to reproduce the same result, which makes us call into doubt all kinds of biological conclusions that we may be taking as axiomatic after some scientific research proves, or a claims to prove that something is true. If we can't reproduce it, is it true? Explain what the problem of reproducibility is in more detail. Do you think that tools like Benchling are going to help solve the reproducibility crisis, or at least alleviate it?

[0:53:33.3] SY: Yeah. I think you did a pretty good job of outlining what the reproducibility crisis is, but to actually go into the mechanics is why this might actually be happening. As you mentioned, this is basically when a scientist, like a team of scientists publish a paper and they have some finding, and then people try to reproduce this finding. They follow the methods that the scientists outline and then they do not get the same result. That is troubling for a number of reasons, but basically it shakes the foundation of the scientific method on which all scientific research is being built, right?

You want to, like as people produce incremental results and publish those in papers, we want to be able to trust them, because you're building this – you're laying the knowledge foundation for research that will depend on your research that you're publishing.

There are a number of reasons that can cause this to happen, and they're all interrelated to the state of science as an industry as a whole. One big factor would just be there's a lot of pressure to publish. It's basically as a scientist, that's how you're measured. The name of the game is publishing papers. You want to publish papers and as it stands right now, you really can only publish a paper if it's a noteworthy result.

If you do a bunch of experiments and found that there is no relationship between X and Y, then that's not that noteworthy. Whereas, if you say like, “Oh, when I increase X, Y goes up as well,” then that is at least somewhat noteworthy and then your people are trying to publish results that are positive as opposed to, “Oh, the state of the world was not changed.”

What this incentivizes, and I'm not saying most scientists are actively being disingenuous in their work. I think even subconsciously what this incentivizes is people to really, really want

results that are meaningful, to the point where it can actually distort the scientific method. An example would be, if you're designing an experiment – suppose you're trying to measure the effect of plants at a university setting. If you have plants, then does that affect people's behavior? Do they smile more? This would be a example of some social science experiment.

If you aren't very clear about how you define the limits of your experiment from the beginning, then you could just literally wait until your data looks good and then publish it. How that might look is you say, okay, we have these two experimental groups. There's one classroom where we decide to add a bunch of plants and we want to see what effect that has and there's another classroom where we just – where we didn't change anything, right? Then over the course of two weeks, we can see oh, people – students are more engaged in the classroom, but maybe that doesn't actually happen. Then maybe what if we wait three weeks? No, we haven't seen any change yet.

What if we wait four weeks? Oh, wait. You notice the change in the behavior in the classroom and they're going to say, “Okay, now let's end the experiment, right?” Then you would see that if you just look at that, then you're basically fishing for the result, because you didn't say upfront when you wanted to stop this experiment. This is a pretty common problem in science. It basically means that because you as a scientist are pressured to find meaningful results, you will be subconsciously incentivized to distort your experiments to the point where you can actually find something.

I can also speak about what the solutions might be and how Benchling could actually play a role in this. I think there is one vision where biological research, or I guess scientific research in general is completely open sourced, right? Right now, compare the biological world, or the scientific world to say Github. Github, like everyone is publishing their work. It's not research about their code right? Leading tech organizations contribute, so industry participates as well. There's just a lot of – there's this spirit of generosity. People want to give back to the open source community, even if you're not actively contributing, you just want to put up a repo and open source it. That's basically the exact opposite in academia and in scientific industries, right?

In academia, you are heavily incentivized to publish only things that really, really matter. After you publish it, it's behind a peer-reviewed journal. These journals are basically gatekeepers,

because not everyone has access to them. What's probably the worst part is that because journals have a limited ability to distribute, like they have limited capacity for what research they want to highlight and they're these prestigious gatekeepers. Journals are incentivized to only put out papers by scientists putting that, or discovering really, really different results.

No one is going to publish your experiment that said like, “Oh, there was no effect between X and Y, because that's just basically not that interesting.” What this means is that you could have, say eight people performing the same experiment, unaware of the other eight teams doing the same experiment. Then if only one of them just by some random chance sees a result just within normal statistical variance, like gets a result, they'll publish it. Then the eight other people who perform the same experiment found that they didn't see a relationship between X and Y, don't publish it. Then now the entire world thinks that there is this relationship.

I would say one big thing that we could do is just find some way to encourage people to actually share their negative results, because do you basically have this selection bias right now where peep or only sharing results that are prestigious, or significant, and that greatly skews and contributes to the reproducibility prices.

I think how a platform like Benchling, or even other organizations could be playing a part of this is basically just bringing the attitude of open source community to science. Mentioning at least, a lot of academics will collaborate across labs, via Benchling, you can have your lab organizations page. On Benchling, it will highlight the papers that your lab has published, it will also mention the actual DNA sequences that your organization has used. Now this isn't directly addressing the problem where people aren't publishing their negative results, but at least it's creating a community like Github where people feel free to collaborate and share their data.

[1:00:07.4] JM: Well, that makes sense. Sherwin, I want to thank you for coming on Software Engineering Daily. It's been really great talking to you.

[1:00:12.2] SY: Yeah, thank you so much. This has been great.

[END OF INTERVIEW]

[1:00:17.4] JM: The octopus, a sea creature known for its intelligence and flexibility. Octopus Deploy, a friendly deployment automation tool for deploying applications like .NET apps, Java apps and more. Ask any developer and they'll tell you that it's never fun pushing code at 5 p.m. on a Friday and then crossing your fingers hoping for the best. We've all been there. We've all done that. That's where Octopus Deploy comes into the picture.

Octopus Deploy is a friendly deployment automation tool taking over where your build or CI server ends. Use Octopus to promote releases on prem or to the cloud. Octopus integrates with your existing build pipeline, TFS and VSTS, Bamboo, Team City and Jenkins. It integrates with AWS, Azure and on-prem environments. You can reliably and repeatedly deploy your .NET and Java apps and more. If you can package it, Octopus can deploy it.

It's quick and easy to install and you can just go to octopus.com to trial Octopus free for 45 days. That's octopus.com, O-C-T-O-P-U-S.com.

[END]