

Jessica Bell:

This is *ACM Bytecast*, a podcast series from the Association for Computing Machinery, the world's largest educational and scientific computing society. We talk to researchers, practitioners, and innovators who are all at the intersection of computing research and practice. They share their experiences, the lessons they've learned, and their own visions for the future of computing. I'm your host, Jessica Bell.

Jessica Bell:

Today, I would like to welcome Theo Schlossnagle, who is a serial entrepreneur, longtime ACM volunteer, and gentleman farmer. He also just happens to be the founder and CTO at Circonus. Welcome to *ACM Bytecast*.

Theo Schlossnagle:

Thank you for having me.

Jessica Bell:

Well, first off for people who aren't already aware of you, can you give us our intro to Theo pitch?

Theo Schlossnagle:

Pitching myself, that's something I'm very good at. I've been doing computing for quite some time. Started my interest in that, I guess when I was in middle school. But ended up going to Johns Hopkins for computer science. Stayed there for far too long. Left with far too little. Started a couple of different companies, beginning with a professional services and consulting company around internet scale growth in 1997 called OmniTI.

Jessica Bell:

Awesome. I'm always curious about why people get into computing. Were you the typical story of, I was taking apart computers when I was young, or did something particularly draw you to that field? What brought you to tech?

Theo Schlossnagle:

I don't know. I guess relatively young, I would say when I was about eight or nine, my parents got me an Apple IIE computer. When I found out that I could program it in basic, I just found it to be really interesting. I draw a lot of parallels between computer programming, which indeed is different than computer science, but a lot of parallels between computer programming and puzzle solving. They use the same portion of the brain and I find it to be stress-relieving to be frustrated in that way. I like doing puzzles. I like writing code. I think they're the same thing.

Jessica Bell:

Interesting. That's such a on point view of code. I remember when I was first learning, I had this little puzzle game where you would snap in the different statements together to think about logic and programming. That's an interesting way of putting it. I like that.

Theo Schlossnagle:

I had to pick one of the things that really drew me into it deeply was, I think, that there's a point at which you understand enough of how computing works to really understand what's going on when you write up program and interpret or compile it to machine code and operate it and run it. After you take in a computer systems architecture class and you know how the chips light up.

Theo Schlossnagle:

I started as an electrical engineer, a lot of gate design, [inaudible 00:03:01] design. I always find it most fascinating when things don't work the way that I intuitively expect them to. I usually get called in to look at kernel bugs and compiler bugs, and regular bugs, and things like that. I find that violating the layers of abstraction that are required to make computers useful and productive tools for people in order to solve problems to be just an absolutely fascinating journey.

Jessica Bell:

Is that journey what ended up leading you into working with architectures and scalability? Talk to us a little bit about that journey from what could have been an academic career path, but went into founding an entrepreneurship path.

Theo Schlossnagle:

I think it was actually an outcome of opportunity more than anything else. I ended up graduating from Hopkins, I think in 97 and then with a master's in 99. Then I pursued a PhD until 2003, but left without it. During that timeframe, as everyone knows, that was when things got pretty exciting on the internet. It was an incredible shortage of expertise in the realm of large scale distributed systems and high-performance code. I had some of that expertise through my research at Hopkins. I was working on distributed systems, things specifically on load balancing algorithms and techniques. It was directly applicable to the large scale web.

Theo Schlossnagle:

As all of my peers in school graduated and went on into industry, when they ran into problems and they were short staffed, they lobbed my name around often as someone to call in as backup.

Theo Schlossnagle:

I ended up founding a company around that and was pulled into some really exciting engagements with some of the biggest internet companies in the world. I got to see a lot of

really interesting problems. Again, it's like looking at a puzzle. Things don't work the way you expect and you need the pieces to fit together differently. It's a great mental exercise that I find really satisfying.

Jessica Bell:

Talk to us a little bit then about your experiences between the beginning of that career, when you're being called in to solve these issues that might be down a couple layers of extraction. Have you seen a major change of what kind of puzzles are happening 10 years ago, 15 years ago, versus now? As the internet grows, as we see a lot more of these very large scale companies, has it changed, or is it essentially the same problems?

Theo Schlossnagle:

I think the problems are very similar. The problems in distributed systems are more prevalent now because most of the systems that we use, if you deploy into the cloud, there's basically no way that you can't have even a traditional distributed system. But even if you're writing an iPhone app your iPhone is a distributed system, it's got all these different processors on it that run at different speeds and different buses, and all sorts of stuff. I think that one of the things that gave me a distinct advantage of being able to apply my skills to industry is that when I started at Hopkins, I ended up working in the computer science lab as a technician or a systems administrator. I would make sure that systems ran, they got installed and upgraded and patched, made sure the students weren't exercising exploits, which they were always trying to do, that sort of thing. A lot of grungy operations work.

Theo Schlossnagle:

When I went to apply computer science principles to problems, it wasn't just an idea. There was the ability to go in and recompile all the apps and change the code and deploy it and understand how those systems, not only were they designed or written, but how they were deployed and operated as well. It allowed me to take my ambitions to the finish line. I think that one of the challenges that I think is fantastically eroding, but one of the original challenges is a lot of people who wrote code didn't know how to provision servers and rack them up and get the networking working. They needed an IT team to be able to enable the deployment and operationalization of their code.

Theo Schlossnagle:

Today with the cloud, this whole no ops movement where you can sign up for an Amazon or a Google account or even a Microsoft Azure account and deploy your code with a couple of clicks has removed those barriers. But I think that it's shortsighted to think that that's how Netflix runs.

Theo Schlossnagle:

Netflix and Google and Microsoft and Amazon, they all hire data center operations people, they hire hardware specialists, a lot of them make their own chips. They also really understand how

the underlying components that drive the services that they use work and they build them. Kernel engineers and things like that.

Theo Schlossnagle:

The world's a much more complicated place. I think that the biggest challenge that people run into today is that when you struggle to solve a problem, when you hit a bug, when you hit unexplained behavior, the layers of abstraction have become more difficult to penetrate. If I have a kernel bug and I'm running in the cloud, I can't see that kernel. It's very difficult to get access to the kernel that's running and the VM or hypervisor bug or things like that. Those are different challenges.

Jessica Bell:

Do you feel like the trade offs in accessibility on writing your code and getting it deployed via these one click or cloud platforms, do you think those trade offs are worth it? Or is it a it depends on your situation question?

Theo Schlossnagle:

Absolutely situational. But I would say that 99% or more of the time it's totally worth it. If you can increase developer productivity and take new ideas to an audience, actually take an idea to an audience for use with less friction. Then that's a huge win. And then again, if you start running into problems where those abstractions are a hindrance, the only reason you'd really run into those, where it matters, is if you have an audience, you have demand for your services or your application or your code, and then that's a problem you want to have. It's great to tackle it at that time.

Jessica Bell:

What things would you say to a company that is beyond the startup stage, not the scrappy little, we're trying to get anything done by the skin of our teeth, but we're ready to scale up a little bit? What are some of the common mistakes that you see or the common questions that people fail to ask themselves at that stage?

Theo Schlossnagle:

Most code and infrastructure and design patterns for architectures of code resemble organizational structure. You should take a really hard look about what your organizational structure looks like, how you think it's going to look in a few years after it grows a little bit more, what you don't like about that, and realize that all of that stuff that you don't like will also manifest itself in your code and your architectural designs.

Theo Schlossnagle:

I think that there are a lot of trends around mono lists versus microservices. Using the best tool for the job is a great example of that as well, where people always want to use the best tool for

the job, but if you have 40 employees who all decide that a different tool is the best tool, then you have no shared domain knowledge over those tools and you have these severe friction impedance mismatches when handing code or doing code reviews. It's an organizational dysfunction, even though it's a local optimization.

Theo Schlossnagle:

A lot of this, again, local optimizations in organizations usually have pretty bad consequences on a macro level. The exact same thing is true for code. I would say that discipline and structure and a focus on making sure that your organization skills are solid, you know where your things are, you know where your code is. Whether or not you use test-driven design, for example, which I personally am not a fan of. Every time somebody argues with me, they go and redefine test-driven design in a way that's not objectionable, but the concept of I'm going to write my tests before I write my code I'm not a huge fan of.

Theo Schlossnagle:

I absolutely think that code must have tests, but you have to know where your code is, where your tests are, what types of coverage you have, where your competence lies. All of that stuff are really organizational skills. Strong organizational skills that have uniform adoption across an organization will prevent you from getting into places that you can no longer get out of it.

Jessica Bell:

Where do you think some of these mismatches come from? Is it a fact of our industry is startup and we'll just do everything as fast as we possibly can without a little bit of intention? Or does it end up being a fracture in the kind of developers or programmers that you're hiring? Talk to me. Where do you see those cracks coming from?

Theo Schlossnagle:

I would say that in immature and pseudo mature organizations, so not talking about Amazons and Googles and things like that, though occasionally they manifest these problems, the concept of the inmates running the asylum is incredibly apt. There are a lot of technical challenges where people feel that they need to hire exceptional programmers for, and who are they to say how they should do their job? If you can imagine having a complicated finance situation where you hired 30 finance people and said, don't tell me how to do the books, you each individually decide what the best way to do the books is. Enjoy your audit. That's going to be a bad situation.

Theo Schlossnagle:

Yet in computing, all of the engineers, definitely not all of the engineers, but a lot of engineers that I've interacted with that manifest these problems feel that they were hired because they know best and they should be entitled to make all of these decisions on their own. That never

makes sense. Especially in a large organization, that lacks any uniformity, which is very dangerous.

Theo Schlossnagle:

I think that, at the end of the day, it really comes down to simply the inmates being allowed to run the asylum. That, again, you want input and you want new ideas. This industry changes ridiculously fast, so you need that infusion. But you need control over that. You need a gated introduction of that stuff. It needs to be carefully deliberated before it's brought into an organization. You don't want 14 different language frameworks in an organization with 20 different people. That's an enormous problem. It's arguable that you may not want 14 in an organization with 500 people.

Jessica Bell:

I'd like to hear a little bit more about your founder experience as well. Do you feel like that is something that just happened out of necessity to find clients to mattress skillset? Or is it something that you were drawn to as part of a entrepreneurial spirit?

Theo Schlossnagle:

I think it's more of the latter. Well, when I first started, there weren't really any of the big tech companies in the way that they look now. If I was starting my career now, I think that having a job at an exciting tech centered company that isn't a tech company. You've got high frequency trading companies. You've got finance companies, like Bank of America things like that, that are not tech companies, but they are incredibly tech centered now.

Theo Schlossnagle:

That old saying that every company is a tech company was absolutely not true in 97 when I started my first company. What I found was that I was really excited about doing technical things, applying computer science concepts, making sure that things were maintainable and operable and things like that, but I didn't want to do it for the sake of computing. I wanted to do it for the sake of other businesses. I found that I was being grossly underpaid to do that at the first company that I worked at, so I flipped and started my own, mainly because I thought I could run a better company that I enjoyed working at more. I feel like I was right.

Jessica Bell:

I was about to ask, do you feel successful in that?

Theo Schlossnagle:

Well, I founded four companies. One of which is closed. Well, I guess I actually shut down the first one last year as well. But they were all relatively successful. Three of them probably combined had 500 or 600 employees across them. I feel like I created a lot of value for customers and for employees, a lot of livelihoods created and got a lot of feedback from not

everybody, but many of the employees that they found that the experience at working in those companies was foundational in their careers. I am very satisfied with the outcomes of that.

Jessica Bell:

Do you think your programming thought process helped or hindered in the entrepreneurial set of starting a company, being a founder? And then do you think, vice versa, does your founding experience help or hinder your programming brain? If that makes sense?

Theo Schlossnagle:

I think that there's a little bit of overlap. Quite honestly, I think that it's very rare to find one person who is all things. A sales and growth marketing person, I am not. While you can grow a company in the beginning on evangelism, most companies, in order to hit a second level of growth, really need to have somebody who is focused on that. Who's focused on growth, focused on marketing, focused on sales, sales process, training, onboarding, all of those things. These are not technical things. They have nothing to do with computer science. They're very business focused. And there's only so much room in your brain.

Theo Schlossnagle:

I would say I'm not very good at those. Being a computer scientist hindered the ability to grow only in that I had to hire someone else. I think it was very important to that self recognition that, hey, look, I'm not good at all of these things. I need to hire people who are better than me.

Theo Schlossnagle:

My fantastic vision of every company is to figure out anything that I'm good at and find somebody better and hire them. I don't want to be the best at anything in my companies. It just means I didn't hire the right people. But I would say that from a founder's perspective on how it positively influenced my engineering, I would say that the customer is always right. While that's not a perfect statement, it is incredibly important centering thesis.

Theo Schlossnagle:

Like the test-driven design thing. It's one of the reasons why I find code testing to be a little misguided in a lot of ways. It doesn't need to be, but tends to be in practice, is that you can write 100 tests to make sure your code works, but the only real test is do I have a satisfied customer that wants to return? It's very hard to write a test to test that, but that is ultimately what needs to be tested when you do a commit and keeping that perspective throughout the entire process of building new products and maintaining old ones has been incredibly valuable.

Jessica Bell:

Right, technology at its core is a people issue, not a machine issue, I feel like.

Theo Schlossnagle:

It doesn't matter if it's fast. It only matters if it serves its purpose and it's useful to people.

Jessica Bell:

Right. What would you say to the group of engineers who are coming out of their technology programs, coming out of jobs at big technology companies and want to go down this same founder path? Do you have any tidbits of wisdom or encouragement or maybe not encouraging at all?

Theo Schlossnagle:

I won't say that people shouldn't do what I did. That would be hypocritical. I would say that there are a lot of aspects that a founding at a company that are very nontechnical and very difficult. There's a lot of stress in the financial aspects of it.

Theo Schlossnagle:

The first three companies that I founded had no venture capital funding. There were liens against my mortgages. If I screwed up, I lost everything. When you hit hard times, and people don't tell you the hard times are not always when you can't make a sale, it's when you've made a sale, you've done the work and you can't get paid. These are really not technical issues at all. These are business issues and then you don't have enough money to pay people, so you have to pay your employees before you pay yourself.

Theo Schlossnagle:

All of these things are stresses that are difficult. You need to be able to cope with stress well., You also can't really talk to your peers in your organization about your challenges. that old saying, it's very lonely at the top, it's very true. you need a network of support that's outside of your company, either through fellow founders, fellow CEOs, or just family and peers that are outside of your direct report chain or industry.

Theo Schlossnagle:

There are a lot of challenges that are around entrepreneurial-ism that are very real and they're very long term. It's a hard process, but I would say that it's an incredibly rewarding thing. When I look back and people ask what I'm most proud of... I ended up founding a company called OmniTI first, and then a company called Message Systems, which is now called SparkPost, and Circonus, and then another one called Font Deck, which is no longer around.

Theo Schlossnagle:

I think SparkPost today, I think delivers probably 30-40% of the world's email traffic. If you use LinkedIn or Twitter or AmEx or your Chase Visa card or whatever, the emails that you get go through that software platform. You might think that I'm very proud of that. I'm actually really proud of the positive feedback I've gotten from so many employees over the year and the



number of livelihoods I've helped create. That, to me, is something that I could not have done if I was a line level employer or even a manager.

Jessica Bell:

I think about that part. Great. Well, in our last couple of minutes, I always like to hear from people a little bit about the future, what you're worried about, what you're thinking about, what you're excited about. I find that very curious for people who have been, especially for people who have been in the industry a lot longer than I have, I'm a relatively new technologist. What do you think about technology at night? What are you super excited about? What are you scared about? Tell us about the future of Theo.

Theo Schlossnagle:

There's so many things. We're really in the midst of an enormous transformation. The fact that I can talk to most of the devices in my house and they actually understand me, even when I mumble, that's truly amazing. If you rewind 15 years, that was unconscionable. That was AI then. We realize that's not AI now. That opens technology and productivity and convenience levels to the world in a way that hasn't been there before.

Theo Schlossnagle:

Things that concern me, there's always this constant backdrop of the ethics of computing. Computing has great power to disenfranchise people just as it has power to enable people. If we're not very careful as engineers and product managers and companies building the software, we are giving tools to manifest the destruction of sections of society, if not society itself.

Theo Schlossnagle:

That sounds very doom and gloomy, and the only reason that I don't think it's doom and gloom is that we really don't want that, but the potential is there, so we really need to be careful and have a consistent adoption and focus on ethical practices in software engineering and software product design as well. That's a constant backdrop.

Theo Schlossnagle:

On the more technical level, I would say that computing is producing machine data in a way that is just ginormous. There's just no word for the size of the machine data that's coming out of systems now. I expect in 10 years your smallest IOT device will be able to produce a gigabit a second of questionably valuable machine data.

Theo Schlossnagle:

The question is, what do we do with that? Where do we put that? How do we change that into value? That's one of the things we focus on at Circonus, the amount of data volume that we've seen when we started the company.

Theo Schlossnagle:

We started the idea of the company probably 15 years ago. We have one times 10 to the maybe 12th or 14th, more data coming in per second now than we did then. We have trillions of samples per second coming in from devices around the world. You have to make sense of that and produce value out of that and do it economically, because that device, sometimes they cost 25 cents and they're producing data that, in the old days, would cost tens of dollars a month to process. There's no economic model around that.

Theo Schlossnagle:

I think that it's a constant leapfrog of technological advancements, business use case advancements, economic advancements, and then there's this thing that I feel like it's been ignored, which is this ethical advancement of should we really do this? What are the consequences of doing this?

Jessica Bell:

Right. Exactly. Great. Well, I don't think that was all doom and gloom. A little bit of both.

Theo Schlossnagle:

Excellent.

Jessica Bell:

Well, it was really nice to talk to you and thank you for all of your insights and recounting on your history. Thank you so much for joining us.

Theo Schlossnagle:

Well, I really enjoyed it. Thank you.

Jessica Bell:

*ACM Bytecast* is a production of the Association for Computing Machinery's Practitioners Board. To learn more about ACM and its activities visit [acm.org](http://acm.org). For more information about this and other episodes, please visit our website at [learning.acm.org](http://learning.acm.org). O-R-G slash B-Y-T-E-C-A-S-T. That's [learning.acm.org/bytecast](http://learning.acm.org/bytecast).