EPISODE 666

[INTRODUCTION]

[0:00:00.3] JM: For most of history, a typical musician would learn to play one specific instrument. As synthesizers became available to the public, it became commonplace for a musician to create their own instruments using hardware and software. By the early 2000s, a digital audio workstation piece of software allowed a musician with a laptop to have access to the tools of a record producer. All these tools changed how music is made. They increased the work of a musician by massive skills of complexity, and this ultimately gave rise to new genres and new ways of creating music.

Creating electronic music on the computer today is a practice much like software engineering. Iteration, modularity and software architecture skills are required to build a song intelligently. Music engineering also requires working at numerous levels of abstraction; the synthesizer level, the song arrangement level, the mixer level and the design of melodies.

Dom Kane is a musician and sound engineer who writes music for Mau5trap, a label started by Deadmou5. He's built software synthesizers. He's worked with numerous artists. He’s a producer. He's written music for film and TV. He joins the show to talk about working as a professional electronic musician.

We also talk about the overlap between engineering and the different facets of creating modern music on the computer. It was great to talk to him and I hope you enjoy the show.

SPONSOR MESSAGE]

[00:01:44] JM: Accenture is hiring software engineers and architects skilled in modern cloud native tech. If you’re looking for a job, check out open opportunities at accenture.com/cloud-native-careers. That’s accenture.com/cloud-native-careers.

Working with over 90% of the Fortune 100 companies, Accenture is creating innovative, cutting-edge applications for the cloud, and they are the number one integrator for Amazon Web

[INTERVIEW]

[00:03:22] JM: Dom Kane, you are a music producer out of the UK. Welcome to Software Engineering Daily.

[00:03:28] DK: Thank you. Thanks for having me.

[00:03:30] JM: We last spoke many years ago, I think three or four years ago, when you came on. It was a podcast I had called the Quoracast, which I think many people listening to the show probably don't know about, but I had met you on Quora and I had read a lot of your answers. You had a very scientific approach to music and music production, music technology, music engineering, and since then I've followed your work. I've read a lot of stuff and watched videos and listened to podcasts that you've done.

When I started this podcast, I wanted to do stuff about software engineering, but also the higher level engineering that happens to involve software. So I think of what you do as engineering with software, but you're not actually building software most of the time. So I think our conversation will probably vacillate between the question of building software and the question of engineering music. But I guess we should just start by level setting for people who have little idea what a producer does. What does your work as a producer typically entail?

[00:04:33] DK: Okay. So my career is kind of split three ways I tend to tell people. So I think music producer first and foremost. So I suppose most people would associate that to being a recording artist. However, rather than using my voice to sing, then I use synthesizers to create my own voices with those I suppose. So that's really first and foremost, my main job, but on top of that I also, because of my experience in history with various synthesizer and synthesizer companies, I'm also a sound designer as well. So I develop a lot of the sounds that if somebody's going to go out and buy a synthesizer, then there's a good chance I made some of the original sound content that comes in those.

Then on top of that, I tend to do a lot of audio engineering as well, and that's I guess perhaps where we overlap somewhat, because although I would never call myself a dev or coder of any kind, I do deal with quite a lot of coders when they're developing the software packages for music producers or varying artists, and I do a lot of the testing for them. So I suppose it's more than the consumer side, but not quite low level stuff. But at the same time, I have a degree in hardware an analog engineering as well. So I guess that kind of all comes into play really. So yeah, fingers in pies.

[00:06:11] JM: So do you say electrical engineering?

[00:06:13] DK: It involved electrical engineering. So it was analog and digital audio system design, which is a bit of a mouthful. So yes, so it involved – For example, we built hardware audio compressors and things like that and we also did digital equivalents as well. Again, it wasn't low-level coding. It was more using the object orientated software packages, like Maximus P. Sometimes it would be sort of numerical stuff like MATLAB to sort of put in your algorithms to sort of try and workout what the software should do in theory, and then obviously get to a point where you can print it to a PCB and then build the actual product.

[00:06:57] JM: I always wondered what the different levels of abstraction were that went into making a hardware synthesizer. I think of a slightly better idea of how it works with a software synthesizer package. But maybe could you contrast the engineering process between those two things, like hardware synthesizer versus a software-based synthesizer?

[00:07:18] DK: I mean, I think when you get to low level stuff, there really isn't that much difference. You're essentially building resistor capacitor circuits, and in the digital realm, rather than building the circuit, I suppose you're instructing the software package to reproduce what the equivalent resistor capacitor circuit would've done. Yeah, I don't know if that answers your question.

But I mean hardware and software really – I guess there's a bit of a gray area with music products and consumer products in the music industry when we're talking things like synthesizers, for example, and lots of them nowadays, they don't tend to rely solely on analog resistor capacitor circuits. A lot of them are actually, I suppose – My mind's drawing a blank now obviously. I would say FGPA, or FG something.

[00:08:13] JM: Yeah. FPG – field programmable gate array.

[00:08:17] DK: That's the one. Essentially, it's a small computer running software, but actually with tangible buttons and dials, and the vast majority of synthesizers now are essentially a software, I suppose, a one-off software package delivered as hardware.

[00:08:35] JM: I actually saw your name come up recently when I got synthesizer pack, which was Serum, and you had programmed some of the synths in it. I think that was what reminded me. I was like, "I haven't talked to Dom in a while. I should reach out to Dom, because I've downloaded this synthesizer." I actually paid for this one. I was flicking through the presets and I saw your name and I was like, "Oh! Dom made this," and it's a great bass instrument and I'm going to use it in this song.

What's the process of programming a synthesizer in one of these – Or, I guess programming a preset in one of these synthesizers?

[00:09:14] DK: So with serum, it was designed and created by Steve Duda, who is just an absolute mathematical genius, and it started – Well, it must have been 2012, something around that, and he sent me what was really just a sandbox sketch of what could have been a synthesizer at the time. I mean, it was really fundamental basics, and he was asking me my thoughts on it, really. So I sort of wrote down some thoughts and he sort of thrust that way. He was wanting it to go and he had sort of explained that he wanted the wave table synthesis.

So, without trying to get too deep into synthesis types, but there are, for example, most common synthesizer used it's what's called subtractive synthesis. So it starts with a noise engine essentially creating white noise and then by tuning the dials and pressing the buttons, you're essentially removing the unwanted bits of noise to create a sound that you actually want.

Whereas Serum was fundamentally based on wave table synthesis, which is where you have a looping waveform that can be any size and shape, and it fixes to a grid system so you can adjust the loop start and end points to create entirely new sounds and then layer them in. So a lot of it started with us recording various sounds. It could have been a drum strike, or it could have been an old vintage analog synthesizer then analyzing the waveforms from there and finding certain pars that would fit a 10-24 sample grid. It was really sort of trial and error at the beginning. Then you'd have a dial where you can sweep which part of that waveform. So the waveform could be, let's say, half a second in length, but you're actually only looping 5 to 10 milliseconds of it to create an entirely new waveform and then it would be sweepable within the larger waveform so that you could manipulate the sound in the synthesizer.

Yeah. So that's kind of how wave table synthesis begins. It was a mammoth project on Steve's behalf and, yeah, the finished product is just incredible. So from my perspective, that's really where it started. It was quite a low level involvement, I suppose. Then as he started developing more graphical features to it, then it got to a point where he had decided, “This is a stable vision of what will eventually be the final product.”

So then it's a case of me going through those wave tables and sounds and really just tinkering with it. For me, personally, what I tend to do is I tend to really just work normally and start writing music and coming up with melodies and ideas in bits and pieces using that synth, and then every now and then I’ll sort of keep tweaking it and tweaking it and then I'll go, “Hang on, this is a great sound,” and I'll save that preset state and then move on to another one and just keep moving on until eventually you end up with a bank of sounds that you know are usable because they work in your own productions.

[00:12:32] JM: That's great way of coming up with synths. If you start with the raw synthesizer and you start working on a song and you have some vision, you develop some vision for the sound overtime and how it fits in with the context of your song, and if it's a good synth in the context of your song, it's probably going to be a good synth in the context of other people's songs.

[00:12:56] DK: Yeah, that’s certainly what I hope will happen. Sometimes you’ll develop a sound that really could just be a background sound to a tract. Obviously, when you buy a new synthesizer, it's jam packed, full of huge sounds that sometimes you can just press one on a keyboard and you think, “Oh my God! This is an entire track in its own right.”

But I think those oddly tend to be the least usable sounds from my perspective as a consumer, because I don't want the synth to do everything for me. So for me personally what I’ll tend to do is start with a bit of a blank canvas and just start writing my own music and start developing little melodies and counter melodies and then maybe in the counter melody I’ll start tweaking something so that it' really becomes more of a faint background sound and it's at that point where I start to go, “Hang on. This is now usable,” and that's where it becomes more of a preset for me.

[00:13:53] JM: So, as you said, Serum is a wave table synthesizer and the primitive sound that is getting manipulate is it's a wave file.

[00:14:04] DK: Yes.

[00:14:04] JM: Like a an actual wave file. So like a wave file that you –

[00:14:08] DK: Actually, Steve took that to a whole new level eventually in the latest version, well, I think probably from version one onwards. He also made it so that if you really delve deep into serum, you can actually draw your own waves. So it's also got essentially a wave table creator in there. So if, for example, you wanted a saw wave, which we all know to look like a saw's tooth, you could use that and then you can switch on the pen tool and then just draw a couple of additional spikes along the harmonics of that wave and create an entirely new sound from that, and it's all very, very usable.

[00:14:49] JM: How does the usability of that wave table contrast with the other wave of building a synthesizer where it starts with white noise?

[00:14:59] DK: I mean, they're just two entirely different beasts I suppose from both a consumer perspective and a designer perspective, but at the same time they produce – The output isn't necessarily completely distinguishable. But I think subtractive synthesis is probably a more logical approach because you start with noise and you're just removing elements of it. So, for example, you'll have an envelope generated with ADSR, which is your attack, decay, sustain and release, and by using that you can just shape the output wave form basically.

With wave table, I guess it's far more open canvas and allows you to quickly shape far more unique signs. I think in an ideal world, the end-user probably shouldn't need to care about which one is which and what they're doing, because hopefully the user-interface is good enough that it just becomes quite a fluid process anyway.

[00:16:02] JM: Speaking of user-interface, my understanding is that Serum was designed to be a more simplified version of Massive, which is another synth with more parameters. Maybe it's not that simplistic, like wasn't the direct design decision. But I have Massive and the other synths that come with complete nine, which is a native instruments package. I got it on Amazon, and it's like a bunch of CDs. So every time I have to reinstall it, I have to use an external CD drive, because my MacBook no longer accepts CDs.

But Massive is this amazing synthesizer, and I had been working on music for a pretty long time before, and I was just using whatever presets FL studio gives you, which are great. There's some great primitives and you could learn a lot just working with primitives like just super basic ones that FL studio gives you and figuring out how to make use of those. But you get to a certain point and you say, "What is out there? What's at the higher end of the budget spectrum?"

Then you get one of these may be kind of pricey, but relative to how much value you can get out of it, not very pricey synth packages, like native instruments and you check out Massive and you just realize, “Oh my gosh! I have been making my life way more difficult than it needs to be.”

Massive is fantastic. It's got a lot of knobs to tune and it's a little bit different in terms of user-interface than serum, and I still use both of them. But can you describe the trade-offs in those – In like user interface design, because I think they're both – Are they both wave tables of synthesizers?

[00:17:46] DK: Yeah, they’re both very similar. I think picking apart the difference, interestingly you said that Serum was more simplified. Actually it's far more complicated than Massive, but I'm glad you said it was simplified, because I think that was really the ultimate goal of Serum, was being able to provide far more complex sounds and complex algorithms, but keeping the user-interface to a usable position. I think that's where really sort of nailed Serum, was you could be a complete novice user of any synthesizer, and you could install Serum, and within a few minutes you’re getting some great sounds out of it and you can work it out. It's not too difficult.

However, the deeper delve in when you start sort of assigning modulators to further modulators to even more modulators, like LFOs and various things like that, and then you can assign those modulators to effects unit, some things hidden away in the back. It can get hugely complicated after a while. I think that was kind of the key to Serum, was that you could – I hate to say, but a one stop shop of synthesizers because there are a lot of original analog recordings that have gone into the wave tables. There is obviously huge amounts of digital synthesis going on, and at the same time there are huge effects units, there’s distortion, there’s reverbs, there’s echoes, there’s chains and chains of effects you can place on these sounds. It kind of has everything.

Again, going back to that user-interface, I think it does come across as perhaps more simplified than Massive. Actually, I would say it’s far more complex. Like I say, you can draw your own waveforms and really start experimenting with it. But at the same time, you don't need to, and that’s not pushed into you, which is what makes it appear so simple.

[SPONSOR MESSAGE]

[00:19:54] JM: DigitalOcean is a reliable, easy to use cloud provider. I've used DigitalOcean for years whenever I want to get an application off the ground quickly, and I've always loved the focus on user experience, the great documentation and the simple user interface. More and more people are finding out about DigitalOcean and realizing that DigitalOcean is perfect for their application workloads.

This year, DigitalOcean is making that even easier with new node types. A $15 flexible droplet that can mix and match different configurations of CPU and RAM to get the perfect amount of resources for your application. There are also CPU optimized droplets, perfect for highly active frontend servers or CICD workloads, and running on the cloud can get expensive, which is why DigitalOcean makes it easy to choose the right size instance. The prices on standard instances have gone down too. You can check out all their new deals by going to do.co/sedaily, and as a bonus to our listeners, you will get $100 in credit to use over 60 days. That's a lot of money to experiment with. You can make a hundred dollars go pretty far on DigitalOcean. You can use the credit for hosting, or infrastructure, and that includes load balancers, object storage. DigitalOcean Spaces is a great new product that provides object storage, of course, computation.

Get your free $100 credit at do.co/sedaily, and thanks to DigitalOcean for being a sponsor. The cofounder of DigitalOcean, Moisey Uretsky, was one of the first people I interviewed, and his interview was really inspirational for me. So I've always thought of DigitalOcean as a pretty inspirational company. So thank you, DigitalOcean.

[INTERVIEW CONTINUED]

[00:22:00] JM: I think what gives the simplicity for me is the FX panel. Honestly, I haven't tweaked synthesizers enough to be really well-versed in what is a simple user interface versus a complex user interface. I am the kind of person that is a broad swath of the target market for these kind of synthesizers, because I'm not a professional musician. I'm somebody who does it in their spare time. So how can I be productive in a shorter span of time, and productive to me means finishing songs. So that means like being effective in the synthesizer.

The FX panel in Serum, it has like larger panels. Whereas in Massive, I feel like I have to like click on this little drop down menu and like scroll to this little thing, and I can't really visualize what this new thing is going to give me. Whereas in Serum, you've got this FX panel with distortion, and reverb, and delay and it's almost like I've got a mixer right in Serum itself on one of the panels.

Do you think that if you're putting yourself in the shoes of a synthesizer designer, do you want to make it so that the sound designer, the musician, doesn't ever have to assign this synthesizer to a mixer panel? Like, theoretically, you could do all the mixing that you needed to do at the synth level and never have to go to the mixer?

[00:23:25] DK: Yes and no. I think we have reached a stage where a lot of synth developers are now starting to almost over-engineer the software. Interestingly, you mentioned native instruments. I used their machine vision one when that came out. I've got to be honest, I hated it for a long time, and still do to an extent, because what they tried doing was – Essentially, it's a piece of hardware connected via USB. The 16 trigger pads on there appears as though it should be fairly simple. There's some infinite rotary dials on there. I think there's 8 or 9 of those, and that's essentially it, and it came with a software package that you could run inside another software package or as a standalone and it allowed you to drag and drop samples from kick drums, snares, whatever drums, percussion sounds, or even apply notation on piano rolls and things like that.

But unfortunately they had built the software as an entire workstation within its own right. So those rotary dials became useful for the dragging and dropping different effects and things like that. But then those effects you had to assign to either individual pads, or groups, or channels, or stems, and there were so many options that it just felt like every time you wanted to tweak a sound slightly, you felt like you had to apply for a government grant to do that. It just took an eternity and you were in and out of menu systems.

For me, I bought the machine thinking, "Well, this would be great for live shows, because I can just bash these buttons and twists these dials and away I go," and that just wasn't the case. So while that package you can quite literally build, you could build an entire track. You've got the whole workflow. You can work in various different time formats, whether it'd be individual strikes of percussion, or individual notes, or loops, or series of loops, and then you can control which series of loops in what order and then randomizing. So you could create an entire track from start to finish that could be half an hour long.

But that lost its simplicity then, and I felt like whoever had designed that had an idea in their mind and then it was almost too many cooks got involved and it just ended up being this mammoth package. I think a lot of synth designers at the moment, I don't know if it's because they're trying to compete with the likes of Massive or Serum or packages like that where they do have stacks of effects and things, but I think we're getting to a point now where a lot of people are starting to sort of really over-engineered a lot of their synths, and sometimes you just want to be able to buy a toy that makes a noise and just play with it. For me, even as a sound designer, I actually want more simplicity in my synthesizers, which is why I quite often refer back to analog synths and use hardware synths in my work, because I like the freedom of restrictions. I know that sounds like it's a bit of a contradiction in terms, but for example, one of my hardware, [inaudible 00:26:48] synths, I refuse to save a preset on it.

Actually, that’s over the last few years has kind of become my signature sound that I use on that synth. For me, I refuse to save it as a preset. I refuse to recall that sound ever, again even though I have it in almost every track I do. I prefer to rebuild it every time, because every time it's just slightly different. For me, I find it really liberating to be able to make a noise and go, “This is the one. I'm going to hit record. Go for it. Play with all the dials in real time. Record all the automations by hand and then that’s it. I'm happy with that recoding. That's the one. Delete,” and then it’s gone forever.

[00:27:34] JM: Yeah. Will, it can be nice to – I do something similar. I always start from a clean slate when I’m making the song level at least. Even though I could – I do something similar in many different cases, I don't find myself saving templates. Maybe I'm just being irrational and I’m not saving myself time, and I should be saving myself time and saving a template of a song. But I do kind of enjoy the process of it. It's like setting the table or doing the boilerplate work. Builds a little bit of momentum, and you’re right, it does very each time a little bit.

So I’ve wanted to do some shows around the engineering of digital audio workstations. I haven't been able to get anybody from FL Studio or Ableton or GarageBand, but I am curious about that, because what I like about digital audio workstations is it seems like these are super durable pieces of software. Some of these workstations have been around for decades. Well, not maybe not decades, but a decade and a half or a decade, and I imagine like managing a piece of code that is that old can get really hard, and yet they still seem to advance even if it's just on the user-interface level, or maybe as a testament to their core code base that they seem to just stand the test of time.

I knows this is not your expertise, but have you met with any people who work at these companies? Do you have a perspective for how engineering works at Image Line, or the Ableton Company?

[00:29:02] JM: Well, interestingly, one of my good friends who's also called Dom is the owner of Bitwig. So he's the guy behind Bitwig Studio, which is the latest and greatest workstation. I don't mind saying I use that in all of my productions. So I've been fully into Bitwig since vision one. So yes, I have spoken to him a lot about it and it's interesting to hear you say that, because actually I was 100% Ableton for production work, and it got to, I think, vision nine where I started noticing there were serious issues with basic things like MIDI routing. I remember talking to some of the staff there and they sort of acknowledged it off the record and saying, "Yeah, we know there's some issues there and it's enough base codes. In order to change, X, Y and Z, would need to really start from scratch again, and that's not going to happen.

Similar things were happening in Pro Tools, which at the time I was using ProTools for all my mixing and mastering. So it's interesting to hear you say that they've sort of stayed on top of it all. Whereas, for me personally, I felt like people weren't staying on top of it, which was one of the reasons why I got involved with beta testing for Bitwig, and it's very similar to Ableton and in fact was started by ex-Ableton staff.

I know there are rumors all over the internet that they hated Ableton or something at that. It's just not true. They were really just excitable developers who said, "Hey, let's start our own and see what we can do," that's exactly what they did and is now flourishing.

Ableton had some fairly severe issues when it came to – There were PDC issues, which is plug-in delay compensation. So if you had a series of plug-ins on one channel that were fairly CPU-intensive because of the architecture of the software, if you were then to automate the gain up and down across time, there would be delays and sometimes you could end up with delays if you know 20, 30, 40, 50 milliseconds, which meant if you were automating something like a hi-hat, there was just no chance of it being in time and it just became a nightmare.

To my knowledge, those issues really couldn't be fixed without scrapping the whole Ableton project and starting all over again for the developers, which I can completely understand. That's not what they wanted to do. Bitwig was – I don't think it came out in the back of that issue, but they certainly went into it knowing what the modern-day producer expects from a workstation. From my perspective, it's leaps and bounds ahead of all the others, because you can build your own modulators and assign pretty much any parameter to any other parameter, and they've got some just phenomenal groundbreaking tools built-in. They're also making parts of it open source as well so you can actually start to develop your own plug-ins inside Bitwig.

So I think the fundamental platform for any workstation when you're dealing with, I guess, two different timelines in software. So you're dealing with the real-time processing and clip arrangement and development compared to the overall arrangements and timeline. Providing you've got all those in place, then I think 10 to 15 years shelf life is in the world of software. I'm sure that's a really long time. Yes. So I think these companies do incredibly well to be able to get it to last as long as it does.

[00:32:52] JM: The other interesting thing about these digital audio workstations is most of them are not really web-based. They're just these big monolithic piece of software that run on your computer, which is the exception by far in software these days. Most of my work, like I've got 50 web browsers open right now, and then I've got GarageBand open just recording this audio, and that's about it. A couple of other small desktop applications, but most things are moving to the web and it'll be interesting to see there are some developments in browser technology. We've done some shows recently that – I don't know if you knew this, but Reason, they manage to port the digital audio workstation to the browser recently because of this new web assembly technology. So we might see more of these coming to the browser.

Whether that matters, I don't know. We'll see. But it is funny that there have been certainly advancements in music production technology, but a lot of it has not really involved the web. I mean, the splice might be the exception. Don't you find that curious that the internet hasn't really impacted the digital audio workstation world as much, or has it?

[00:34:04] DK: I guess yes and no. So, for example – I mean, pre-internet days, there was a company called Loopmasters who are still going now and I still do regular work for them. As a music producer, if you wanted a new bank of drum sounds or loops or something to play with, then you would go on to the Loopmasters. I think it was a magazine that the style or a catalog, should say, and then eventually a website where you could order a CD and you'd – So there could be some real drummer recordings or it could be some electronic drum machine sounds of whatever and you'd be able to order a CD and it would arrive by post and, hey presto, you've got a new bank of sounds.

Then obviously with the birth of the internet and server space becoming a more realistic thing, it's all downloadable now. Whereas now, they've launched Loopcloud. With Loopcloud, you run it as a plug-in in your workstation, but it's essentially an online browser for their entire catalog, so you can then – I don't know if they do a subscription or something like that. I think where basically you – It's running as a little browser. Rather than browsing your hard drive when you are searching for a specific sound, it'll brose their online catalog and make it available for that project file.

So I think there are companies doing a lot. In fact, I'm working on a project myself at the moment with a couple of friends. We're sort of testing the processing capabilities of a server at the moment to try and offer online instant mastering. So we've sort of had to delve into that. I think I feel like that's perhaps where the industry is heading as well, because with the internet and the availability of all these tools now, I guess the advancement of technology produces traditionally needed an entire studio full of analog equipment and an engineer to run it'll. Whereas now it's all done, I guess, a budget laptop will do these days. That means there are a lot of producers writing their own material in hotel rooms around the world before they go to a gig. They're looking to get that track mustered so it's at least safe to play in their gig later that evening.

So, funnily enough, I think this all came from a Quora question. Somebody asked, "Is it possible to have an algorithm-based piece of mastering software?" and I think that was probably one of the last questions I ever answered on Quora before I then sat there after going, “Well, theoretically, it’s possible. I suppose there is a specific science to mastering music, and I suppose it needs to achieve a certain amount of things and it needs to run through certain checks.” Then I started thinking, “I wonder if that is possible, and can I do it?”

So I then got in touch with a couple of mates who computer science specialists and, yeah, we’ve spent the last couple of years testing out bits and pieces now and we’re hopefully going to get to a stage later this year where we can actually open it up for beta testing.

[00:37:23] JM: Oh, wow! I will be the first to sign up for that.

[00:37:28] DK: That would be great.

[00:37:28] JM: There’s a tool that offers this for podcasting that's really popular, and I can look it up later on and give that to you. Maybe that would be useful. But I found it to be a little too – I'm sure you’re encountering this, where just like there's so many edge cases and it’s like, “What is the general piece of wisdom that you can try to get your software to convey?”

I mean, the thing is you've written about this so much. I remember reading your blog a couple of years ago when I was working on an album and I was at the phase where I was trying to mix and master it, and I was reading your tips and tricks about, “Okay, here's the kind of EQ you want to put on a snare channel. Here's a kind of EQ you want to put on hi-hats. Don't do this with vocals,” and it was really useful, but there's a lot of edge cases.

So with this piece of software, are you going to have to put in all the different channels or is it just going to be you give it the one overall wave file and it tweaks it as necessary?

[00:38:36] JM: Yeah. It would be the one overall wave file. So the idea is, is that a mastering engineer will always say that they can only ever be as good as the mix engineer. Clearly, if you provide a mastering engineer with a terrible mix, then you’re going to get a terrible master. It’s as simple as that. I think the whole point of this project is going to be – So we’re trying to make it learn, and I think that's probably one of the hardest parts.

But eventually it will hopefully get to a stage where the more users use it, the more it's going to recognize what the desired outcome is. So clearly, without giving away industry secrets, there’re lots of statistical analysis within the numbers that will essentially go right. In this genre of music, they tend to prefer this frequency range to appear at this volume relative to this frequency. So it needs to spit out a lot of numbers and then check the numbers and then adjust accordingly.

But, yeah, it's kind of a passion project at the moment. I'm not expecting it to replace real people or anything like that, but I do feel like at some point in the future people probably will be replaced by far better systems than this one.

So I think going back to your previous question about the internet and things being web-based, I think we probably are getting there very slowly. Again, with workstations, the processing power needed for a good workstation to do its job properly, there’s such a combination between things being RAM-intensive and CPU-intensive that really at the moment server aside that have to be beasts of machines working on it to get a workstation to do what we really want.

I think for the basics to happen could be fairly straightforward for any workstation developer. Then would it be in their interests to provide a system that’s really quite restricted and limited when a quick download and you've got your own processing power? I think technology is probably advancing the technology of laptop power is advancing far quicker than the technology of server-side power.

[00:40:57] JM: Going back to the mastering tool, it seems like you could also do the mix level. So let's say I can standardize most of what my channels are. So I’ve a base. Maybe I’ve got two different kinds of bases. I’ve got kind of a wet [inaudible 00:41:15] base and then I've got like a dry saw base. I’ve got a kick channel. I’ve got a snare channel. I’ve got a hi-hat channel. It seems like you could also – You could have a different piece of software that puts those individual things into some kind of set of algorithms and it takes care of that for you. Have you thought about that problem at all?

[00:41:38] DK: It's come up and then quickly being shut down at this point in particular. So the main guy I’m working on this with one of my friends since birth, and we used to run a company together that was for musicians, or record labels I should say, to deliver promo copies of their forthcoming releases to their mailing lists so that they could get feedback from DJs or magazines or whatever. It's probably commonplace now, but back then, this is 2000 I think it was. There wasn't really such a system, and what happened was I was writing music, sending it out to DJs. I remember one time I had written a remix of a track and I had sent it out to probably 50 DJs that I had a list of their email addresses that I regularly sent workout to. Off the 50, you send it to maybe two or three of them would get back to you going, “Love this,” or Hate this,” or whatever. Giving some sort of constructive feedback.

Clearly if it's a big-name DJ that's playing your music, you want to use that in your promotional materials so when it goes to stores, you can say, “As supported by …” Done a remix, and I sent it out to probably 50, maybe 100 max DJs and didn't hear anything back or nothing out of the ordinary. It was probably six months after the release, I found out David Guetta had been playing it on his radio show, and I just thought, “That's huge!” but I didn't know, and I’ve only just found out. So it’s incredibly frustrating then, because I thought, “Well, hang on. If I’d have gone to the online stores and said, “This is being supported by David Guetta, global megastar, I'm sure they’ve made a song and dance about it and they’ve had given it some front page features. It would've been a big deal.

Actually, the sales were not much different to any other release I’d had at the time. So we ended up building this system. We really just sort of built it for myself just as, again, a passion project. What we’ve done is we basically built so that I could upload my mailing list much like MailChimp or something like that, but then also attach the MP3s and then it would spit out a preview file and you wouldn't be able to download those until you filled in the feedback box. So that was basically the premise was, “So if they didn't want it, then close browser. Pend off, move on. If they did want to it, then they’d have to give me some feedback.”

Yes, so we started doing that, and then we decided, “Hang on, this is potentially something.” So we opened it out to the public and set up I think it was a subscription service, and that did fairly well, although neither of us really knew what we were doing in the dev side of things. It was all PHP and an absolute nightmare to maintain. So we ended up giving up on all of that just purely because it was just too much of a headache to run.

I’ve forgotten your original question.

[00:44:57] JM: Well, I was asking about if you could do this at the individual instrument level, if you could say –

[00:45:01] DK: Oh, that was it, sorry. Yeah. So off the back of that we'd sort of learned so many lessons, and I hope half your listeners don't get offended and I hope half your listeners probably at least agree. But the end user of websites can often be just dumbfounding, and some of the issues we had in running that where there were just so many face palm moments where somebody would have a problem that really just wasn't a problem and they'd make a huge song and dance about it, and simple things like the artwork needed to be a square image because it needed to fit the format of the page. Fairly simple, and it needed to be either JPEG, or PNG, or whatever. We'd restricted it to say four or five file types.

Now, you'd think that that would be a fairly straightforward ask of an end-user, but it just wasn't the case. There were people coming back saying, "Oh! It won't upload my image," and we'd go, "Yeah, that's because it's not square, or that's because you're trying to upload an MP3 and that's not an image."

So we did briefly discussed the idea of having it trying to automate mixes, but I think musically and scientifically I think it's probably far too complicated to get into that at this stage. But first and foremost, I think if you had a channel called kick drums, and kick drums are essentially the frontline and focus of any electronic music track, we know full well that people will be uploading vocals in the kick channel and kicks in the vocal channel and it's asking for a nightmaring customer.

Yes. So like I say, that did come up. That's a hugely long-winded answer. I apologize, but it did come up, but it quickly got shut down for those reason. But having said that, the caveat I guess is it wouldn't surprise me in the next 5 to 10 years if that does become something that we'll see. Whether that comes from me or not is anyone's guess.

[SPONSOR MESSAGE]

[00:47:23] JM: Citus Data can scale your PostgreS database horizontally. For many of you, your PostgreS database is the heart of your application. You chose PostgreS because you trust it. After all, PostgreS is battle tested, trustworthy database software, but are you spending more and more time dealing with scalability issues? Citus distributes your data and your queries across multiple nodes. Are your queries getting slow? Citus can parallelize your SQL queries across multiple nodes dramatically speeding them up and giving you much lower latency.

Are you worried about hitting the limits of single node PostgreS and not being able to grow your app or having to spend your time on database infrastructure instead of creating new features for you application? Available as open source as a database as a service and as enterprise software, Citus makes it simple to shard PostgreS. Go to citusdata.com/sedaily to learn more about how Citus transforms PostgreS into a distributed database. That's citusdata.com/sedaily, citusdata.com/sedaily.

Get back the time that you're spending on database operations. Companies like Algolia, Prosperworks and Cisco are all using Citus so they no longer have to worry about scaling their database. Try it yourself at citusdata.com/sedaily. That's citusdata.com/sedaily. Thank you to Citus Data for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:49:09] JM: I think the average musician is just becoming more technically competent, and I think over time – I mean, the thing that gets me working on music is I constantly feel like my productivity is being underutilized and I feel like it's partially my fault, because I get distracted and focus too much on one side of things. Then I realized three hours have passed and I've been working on tweaking this EQ channel on a synth that I'm not even using anymore. That's a problem that's going to be hard for software to fix.

But there are other problems that like software should probably be fixing for me. I do feel like when I read your instructions, which are like you should make your EQ like this. I'm like, "Why isn't this taken care of for me? Why hasn't this been turned into a piece of software?" I mean, maybe to some extent it has. Maybe there are better EQs that I'm not using that have better presets and I'm just not like using those intelligently.

So, I interviewed the CTO of that recently and it was really thought-provoking for me.

[00:50:18] DK: What's his name? Sorry.

[00:50:20] JM: His name is Matt Aimonetti.

[00:50:22] DK: Oh, okay. Okay. It’s not the one in know.

[00:50:23] JM: Yeah.

[00:50:24] DK: Yeah.

[00:50:25] JM: So what’s interesting about Splice is I think they quickly – When they started building stuff for Splice, they were like, “Okay, we are going to do version control for songs and we’re going to do backups and make it easy to go back to a previous version as a song,” which that’s a really useful thing. Then they realized, “Okay, you can also share this with other people, and then you can make it easier to do remixes and fork a song and go off and do your own version of it. Then you've got all the stems,” and they reverse engineer the file formats of Ableton and FL Studio and the digital audio workstations, which was not easy. That in itself was really interesting, but then they found that where the actual money was was in this same kind of thing, the loop – What you’ve said, with the Loopmasters, where people want a subscription to getting these sounds and working on these sounds, and Steve Duda is the Serum side of things. They built a rent-to-own synthesizer model, which is pretty sweet. You don't have to pay a hundred bucks upfront. You pay $10 for 10 months and then you get the synth. That's pretty great for a lot of people.

But the thing is I think what they found is there are so many opportunities that they are focused on certain subset of them right now. The thing that I always am thinking about is why isn’t electronic music collaboration easier? I wonder if it's a facet of the technology or if there is just not a world in which – So like in software, you have hundreds or thousands of developers that will sometimes collaborate on a piece of software. But you don't see this in the music world where there's not a – I can't think of a recent song where there are hundreds or thousands of people who've collaborate on this song across the internet, and it seems strange to me.

[00:52:09] DK: I think it's probably down to the human side. As musicians, we're so self-involved and have such big egos that we want to be in the spotlight. I think it's probably not so much the technology, but more the egos involved. I think, if I'm completely honest, that's probably what it comes down to.

I guess Splice, interestingly, I tested the beta version of that from very early days back in – It must've been 2013, maybe 14, and it was sort of sold to me as a collaboration thing, and I tested it and it worked and it was fine. Then I kind of felt like, "Okay, I'm not sure when I would use this," because – I mean, first and foremost, most of the producers that I was doing any collaborations with weren't using Ableton or didn't have Splice, and I suppose that's probably a different story now. So there were certainly those as fundamental issues.

But I think when you're dealing with – Sometimes I forget that music is actually a creative thing, and I suppose when you developers – I mean, clearly, you're being creative with code. So I don't want to take away anything from that. But you have very clear set definitions and goals that can be perhaps achieved by different approaches, but you're quite restricted in which avenue you can take to achieve that outcome. Whereas in the world of music, there are no bounds to how anything is achieved, and there's no – Because the output is entirely subjective to anyone, I guess it's probably quite difficult to put those restrictions in place, because most musicians don't want any restrictions and they certainly don't want guidelines or tracks to run in one particular direction.

So I think it's probably more a question of humans being collaborative, rather than the technology not providing the platform for those humans to become collaborative, if that makes sense.

[00:54:26] JM: Yeah. Well, those same questions go through my head, because there are a few examples that I care occasionally, where there is just an example of music collaboration over the internet that works out really well. The main one that comes to mind is kind of primitive at this point, but Postal Service. That kind of pop band Postal Service from a while ago, they were big at least in the states for a while, and they were called Postal Service because this was actually way back. They wrote the song over email. This was the lead singer of Death Cab for Cutie and some electronic musician you worked with and a couple of vocalists, and they would just trade emails back-and-forth and pass the song back and forth over email and say, "Hey, I got this. Now, why don't you write your thing," and then somebody else passed it back and said, "Okay, now you write your thing." I was like, "It seems productive," and they managed to be productive. I don't know what magical chemistry made it work.

You hear this sometimes with very – A well-established couple of DJs, or a couple of DJs and a vocalist can work together remotely maybe because they all have high expectations for themselves and for each other and they know that each person is going to fulfill their responsibilities. I've worked with people sometimes where you – I actually worked with you on my last album when I was like, "Hey, I need some help mixing. Can you help me with the mixing?" You said, "Sure. Can you give me very clear instructions for what you need from me?" and I gave you I think some reasonably clear instructions or maybe some vague instructions and then I had – If I gave vague instructions, then I didn't have super clear expectations. I just said, "Hey, I trust you. You run with it and do that." I don't know.

To me, this seem like solvable problems. It seemed like problems worth solving, because I look at software engineering and I look at music and I see two worlds where in both worlds you want and end product that is good. There's some goodness that you can – Maybe it's subjective, but you can still come to an agreement that a song is good or a piece of software is good. I don't know how to get there, but I'm kind of inspired by that vision.

[00:56:46] DK: Yeah, I guess maybe this is where humans haven't caught up with the technology, rather than the technology haven't caught up with the humans. But I find, for me, Dropbox has been an absolute savior. Being able to – So, for example, I mentioned earlier on I use Bitwig as my workstation, and all of my Bitwig project files are contained within Dropbox.

So every time I hit [inaudible 00:57:13], it's uploaded a new vision to Dropbox, which then allows me to go back through previous versions as well, but also allows me to work essentially in shifts with other producers when they also use Bitwig. We make sure that we can maybe set the restrictions of, "Right. Well, I am going to be using these three synths. So make sure you have a version of it as well.” So we’ll make sure we’re running the same versions of each software package. Then we can quite easily just go, “Right. I’ve just made a new version. Go check it out,” and there it is on their computer.

So I think the technology is there for anyone and everyone to collaborate, but I think maybe we’re kind of sticklers for our old ways of working, and I think it's probably more a case of humans haven't quite caught up with the technology that surround them.

[00:58:06] JM: How has your work as a producer change since we last spoke? It's been about four years. I mean, you mentioned a new software project you're working on. What else has changed?

[00:58:18] DK: Good question. I think the last time I spoke to you, I had probably just signed my first track to mau5trap, and I've been releasing with them since. I've also started doing more work for sync companies as well for doing TV and film stuff. Beyond that, I'm not sure what’s changed really. I mean, anything and everything and nothing I suppose.

[00:58:46] JM: Did you build a studio? I feel like I saw some videos on Facebook of you building a studio.

[00:58:52] DK: I did, and now I've lost it all as well. I say lost. I moved on from that. So I was living in Cardiff and I'm not living in Manchester. I’ve just moved here a month and a half, two months ago. So I'm in the process of designing a new studio to build in the garden of the house. Yeah, I had built a studio that was sort of focused on music production, sound design in an industrial state in Cardiff, which I was thoroughly happy in. Yeah, I'm doing it all over again.

[00:59:26] JM: Well, I’ve made my over investments if – No. You don’t have to go into that. But I’ve made my overinvestments in the past, and I can understand. So the work on like TV and film and releasing your stuff through mau5trap. Can you talk a little bit more about that?

[00:59:47] DK: What do you want to know?

[00:59:48] JM: So when you work with TV and film, what is that look like? Do you sit down in front of an episode of House of Cards and then compose the music for it or something like that?

[00:59:58] DK: No. No. No. No. Well, I mean, I suppose that's perhaps one of the ultimate aims. I think it's always been a dream of mine and probably most musicians to be able to provide a full – To do the whole Hans Zimmer. But at the moment, the way it's been working for me is – So I got approached by Universal Music to write some electronic dance music for a couple of TV shows. It was to be used as background music. So it wasn't big money project or anything like that. It was a fairly small project to be used on the backend of a couple of TV shows, which I did and thoroughly enjoyed, because it was such a specific brief that – I can't remember what the brief said, but at the time, it was just so specific. They wanted certain types of sounds. They wanted to certain sort of melodic construct, and I ended up doing that.

I just really enjoyed working to a brief. It kind of made sense for me. So, yes. So I ended up becoming a registered composer and writer for Universal and I've been doing a fair bit of that over the last few years. So what they tend to do is rather than, for example, Hans Zimmer's approach would be direct approaching him and asking to complete the entire score for a film. So this is an entirely different ballgame. So Universal put together essentially catalogs of music that then get pitched towards certain shows, or they might be approached by a show looking for something that isn't in their catalog and then they'll quickly runaround all their writers and composes trying to fill in that gap in the catalog basically.

Yeah. So it's been a really interesting experience, and that's actually then led me on to a few more recent projects, where I've been approached asking, "Have you got something along the lines of X, Y, and Z? We're looking for something for a TV advert." Again, their briefs are so specific and they're looking for maximum 30 seconds of high impact music. It's an absolute pleasure to actually write that I guess because it takes me out of my self-questioning creative head as a music writer and producer. When you're producing for yourself as an artist, you constantly second-guessing everything and overanalyzing your own work. Whereas when somebody slaps down a brief and goes, "We need this." You can sort of just go, "Right. Forget myself, and I'll just go straight for that." Yeah, it can be a pleasure to do.

[01:02:47] JM: I'm totally with you. I mean, I do – Some of the work for this show is interviewing companies and I get paid to interview companies about their product sometimes. I try to do an advertising company a couple of years ago, where I worked with some companies to make advertisements for them with a group of creative people. That's kind of fun too, like working on marketing related stuff. It does impose some of those creative constraints and it can be kind of interesting in that regard.

I heard a podcast recently about – It was Alec Baldwin. So Alec Baldwin has a podcast. It's actually really good, but he interviewed this guy who collects these industrial orchestrals I think is what they were called, but there was a time 30, 40, 50 years ago when companies like General Motors would make entire musicals about General Motors. It was this really weird industrial commercial excursion that a bunch of companies went on and they would just pay these composers tons of money to make these like 30 to 40 minute musicals about General Motors. Really strange time in music/creative history, but I think some of the composers completely loved the work for whatever reason, the constraints or whatever element.

[01:04:12] DK: The money.

[01:04:14] JM: They money, yeah. It doesn't hurt to be paid. If you get paid and they impose some constraints, you're like, "I can be creative with those constraints, and I get paid for it. I'm totally fine with that.

[01:04:25] DK: Yeah. Why not? Yeah. I mean, at the end of the day, I think music can be both an art and a business and there's a huge overlap in there as well. It can be – Those things aren't mutually exclusive nor inclusive. So yeah, I think, because as an artist releasing stuff under my own name, I'm very conscious that I am the brand and therefore I need to focus on particular areas of a genre, or some sort of stylistically it needs to represent me and my brand I suppose. But when you're working under a different name or no name and somebody slaps a brief that says something, "In no uncertain terms, this is what we're looking for." It can be seen as an enjoyable challenge then, because you really don't need to think about it yourself and you can just do it and that can be – I think because I spend most of my time second-guessing my own work, it can be quite liberating to sometimes not have to do that.

[01:05:39] JM: Dom Kane, really great talking to you, and I look forward to working with you in the future.

[01:05:45] DK: Definitely. Thanks for having me

[END OF INTERVIEW]

[01:05:50] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use, and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plug-ins. Use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on-the-fly. GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on GoCD with the new Kubernetes integrations. You can check it out for yourself at gocd.org/sedaily.

Thank you so much to ThoughtWorks for being a longtime sponsor of Software Engineering Daily. We are proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]