**How to Check Your Tests Without Running Them - Kwo Ding @ OQConf 2020**

[00:00:00] **Tristan Lombard:** [00:00:00] We have the one, the only Kwo Ding. How did on today's topic, how to check your tasks without running them?

[00:00:16] This is going to be a good talk y'all yes. Yes. So without further ado, I think friend take it away.

[00:00:23] **Kwo Ding:** [00:00:23] Today I'm gonna talk about, uh, how to check your tests without running them. So I'm really focusing on the steady coats analysis or linting, I'll tell you more about it in a minute.

[00:00:35] Um, but first, well, I got a great, uh, introduction from a, from Theresa. Um, but that's a quick summary. Yeah. I've over 13 years of experience. I focused on implementing test strategies at different immunizations. So basically I go to different clients and see what I can approve their testing strategies. So test automation, strategies, um, and designing test architecture.

[00:01:00] [00:01:00] I specialized in the automation of web mobile and API testing. And I'm based in the Netherlands. Here's some contact information. So don't hesitate to contact me, just use this information, but first, um, yeah. What is lending? So lending is the process of running a program that'll analyze code for potential errors.

[00:01:21] So really detecting the desired piece of code. It's also not a static code analysis. We compete competed has some nice, a nice description, a length or a linter is a tool that analyze the source code to flag programming, errors, bugs, stylistic errors, and suspicious constraints. So really you're looking for potatoes box or, or code that may not Brian, or may not run properly or will not work properly.

[00:01:47] And without running, uh, the actual, uh, application. And the term lending is actually derived from a form shift wall. Actually don't ask me why, but it's a, it can [00:02:00] be from there. Um, uh, it's about, uh, removing unnecessary bits of floss from Shitsville. So that's linting but why do we link? So a lot of time, a lot of times I hear people saying, well, I find old coding style or a year after some examples.

[00:02:19] These are actual quotes from people I worked with. Like I disliked semi-colons at the end of the line for JavaScript, or do you curly brace? I've read so many Twitter threats about the opening call currently brace I'm on a new line or not for better readability. And of course these are all opinions and coders and pinning at it and it's fine.

[00:02:43] But if you want to work together, And work together, uh, effectively, um, to get distracted by all this coding styles is, is, uh, well, not very efficient. So LinkedIn can help, uh, help you with this.

[00:02:59] Well, [00:03:00] this conference is all about the open stores. So here we have a really cool open source project called, uh, popular convention. Um, all they basically does is scans, uh, all these get up projects, uh, and soaps those projects. And it looks for a popular coding conventions. So in this toggle focused on JavaScript because later on, I will show you my new, uh,

[00:03:23] A plugin, which is based on JavaScript and TypeScript. Um, so here are also some examples in JavaScript you can find similar yeah. Examples for different program languages. So here you see the, uh, indentation, uh, so, uh, on the top you see the space. And a bottom of the tab. Um, so if you press the tab key on your, uh, on your keyboard, um, then, uh, you know, insert spaces or an actual tap character.

[00:03:55] And, and if you look at the search community you see for [00:04:00] JavaScript, uh, um, uh, the spaces are heavily used. So if you press the type character, it's a tab, key spaces will be inserted on the line. This is based on this 3 billion guy commits. So is there any nice project to see what, what people are doing actually, uh, you know, for social projects and art, a one, um, the will no space in one space.

[00:04:22] And so it's actually function name, uh, followed by depresses and with a space or no space. Uh, and this one, the, um, uh, the notice space is it's, uh, it's preferred. Um, number one, uh, especially in JavaScript, you see the single quotes and double quotes, uh, the results from this, I was, I found really surprising because in practice, in my experience, I see more single quotes that usage, uh, that double quotes.

[00:04:58] And here you see it's, it's [00:05:00] almost 50, 50. Uh, well, it's leaning towards, uh, what's leaning to a single clouds, but, uh, she can see here in help source world. Uh, yeah, double quotes are also, uh, heavily used in open source projects. Um, and then we, we have coach styles or coach style standardization from different, uh, organizations.

[00:05:24] Uh, I listed here the most popular ones. So, uh, for, for JavaScript, which is, uh, the ones from Airbnb, Google and center JS, uh, what I all have in common is that, uh, well, two spaces for tad single quotes, and they prefer to console VAR and no trading spaces. Um, and unless there's some of the differences, so here she reached out, pulled the semicolons and to training Kamaz, um, That is required by MBB and Google, but not for synergy as, so how, [00:06:00] how it's used today.

[00:06:00] Usually use this, um, you choose a code style like this, and then you can, uh, customized specific rules for organization. For example, I've seen examples that organizations take the Airbnb cut style. And just take a little customized the tab, not to have two spaces, but four spaces. Um, but where does standardization really helps?

[00:06:27] Uh, in, in, in communicating as well. So like I said, uh, why we, Lance is for standardization code conventions and also. It helps with readability and maintainability because you're not distracted with all these, uh, different kind of coach styles. Um, there's a pre code review, so you don't have to do this manually.

[00:06:50] Uh, you know, code review is already been checked by, uh, by linter and last but not least, uh, finding syntax errors box. So actually [00:07:00] a piece of code that might not work or will not work. And this is all again when I was executing the code. Um, so if you look at an application and you see this graph, as in, uh, if you have to solve a bug in production, the customer much higher to solve that, opposed to solving a bog while you are writing the code or writing the application.

[00:07:26] So what does this mean for Tesco? So, um, especially for your eye tests, And this is what we're focusing on today. Um, if can contribute to less flaky tests, I can check for better locators or a check your wedding strategies and also maintainability. Uh, you can, uh, say, okay, we're following your structure. Um, and they may experience, so the collaboration with, with developers is I think one that's the most important one.

[00:07:57] Um, uh, so within your team, [00:08:00] what are, these are fellow there's a lot of developers who also write tests, uh, really talking to the same language instead of having a developer opening up your test code for first time and see all these invitations. Everywhere curly braces on different lines and gets distracted by, uh, all these, uh, layout issues.

[00:08:23] Instead of, uh, focusing on the actual problem was solving your, or your challenge, and then the new condition integration and the continuous delivery world. We, we require rep feedback and lending is I actually one of the. Uh, fastest feedback you can get, because while you're writing the test, you already can get the, get the feedback and the, uh, the same graph as we just signed this in the previous slide.

[00:08:51] Uh, so for tests, uh, production is kind of like running the tests, right. You know, usually in the pipeline. So if you want to, uh, solve a [00:09:00] buck in your test, Uh, on the pipeline, it, it, uh, lots of the customer tired then, uh, if you can already see it when you are, um, uh, writing your test.

[00:09:14] So how does this all work? So basically I literally will take the source code. So your test, right. And it will break down into an abstract syntax tree. Um, and, uh, you have to own these rules. So for example, a test shoots contain container assertion, or we don't want hard, hard weights or implicit the weights, stuff like that.

[00:09:38] Uh, and it will, uh, validate it against it. And the gift you have the result, and this is, this happens instantly and they'll show you how it works with an example. So you have this great website called AST explorer.net. I've opened here. Um, this is also based on a very nice, uh, social project. [00:10:00] So here, um, We have some sample code and here on the right, you see the abstract syntax tree.

[00:10:09] So for example, here, you see the tips is you say a color expression here, and then as you can see, it's a tree of nodes here. She amended it, a member expression and the object names called tips. And you can just click and see. What it is actually how it's broken down, um, in this, uh, in this code. So I prepared a little example.

[00:10:38] So let's say, uh, this is an example of a custom linting room, uh, which you can, uh, create by yourself. So here, if you have the open quality cough, right. And then. I just say up

[00:11:00] [00:11:00] and they'll say, go and stop. So if, uh, if you say, okay, I have some, uh, function,

[00:11:13] then it's actually say, okay, uh, the rules fire, as in it found an error. So if you choose to start the roads not fired. So actually there's this very simple custom, um, linting rail with check whether, uh, this, this code is, uh, is okay. So of course we don't ever want to stop this open Quilty conference, right?

[00:11:40] So this is a, a quick example of how the whole, all of this magic works together that do the presentation.

[00:11:53] so how does it work? You work, uh, you work flow, so a live feedback, white coat and your code there. [00:12:00] Um, we'll run it from the command line and, and because you can run the, from the command line, you can integrate it in your, uh, build process or in your, uh, continuous integration pipeline.

[00:12:16] And, and I don't want to dive into a demo of DS is lean to plugging. I wrote for UI testing frameworks, um, and how you started this actually, uh, I took my drive REO as an example, and I started to write some, uh, some linting tools. Uh, lifting rules, sorry. And, uh, along the way, I discovered that many of these, uh, many of these rules can also apply to different, uh, automation tools.

[00:12:46] So that's how I, uh, uh, uh, expanded that to, uh, support, uh, also, uh, Cyprus playwright, puck deer, and, um, the Tesco cafe. Um, so what I'll [00:13:00] do is go through each of these roles and see some hands on examples.

[00:13:09] Let me switch to this page. So, um, so I might get up, you have this Eastland plug and UI testing, uh, with home, how to install it and how to use it with some examples. So I prepared, um, uh, this project. Um, this is actually a bullet boiler plate project, uh, on the web drive area. I just upgraded, I just climbed it and upgraded to a Provo six and installed the a.

[00:13:44] Yes, lean plugin, UI testing. Um, so if you have use lands installed here, um, you have this, uh, configuration file and basically what you have to do. This is in the written me. I'll just do it hands on. So you add the plugin. [00:14:00] Do I

[00:14:05] Thing. And then what drive area, because this is what drive El Kroger, and we want to, uh, UI testing in here. Um, so now it's actually already enabled by all the default settings. So for example, here, you Fe example tests. Um, here you see, uh, Diaz, lint UI testing. Uh, so you see a void. Absolutely. We're all used to base your relevant project conflict.

[00:14:35] And then our a one, uh, avoid Bravo or page commencing tests. Right. And move them to a, uh, other obstruction layer, like a like page objects. So basically, um, if you move here where you show this to no, absolutely URL. Uh, with the HTTP, it's an absolute URL. Um, it's, it's recommended to use a project conflict, which is, uh, [00:15:00] for, uh, what's available here in the, uh, led dryer, uh, w the IO con uh, file where I can add the, uh, debase URL right here.

[00:15:13] And. Just somewhat something to be. Claire is here. You see for this, no absolute your outside person would prevail because Cyprus also has a, uh, base conflict file, uh, where you can set a base URL for the other, uh, tools. Uh, you do not have that. Uh, well, at least not out of the box. That's why it's a, it's supported only for Cyprus and WebDriver IO.

[00:15:38] If C recommended is actually applicable for all automation tools, um, A supportive, so missing assertion. So here we have an expect right then expect is in assertion. So if I comment that one out immediately, you see high missing assertion tests, even till today, when I [00:16:00] go to an organization from time to time, I see different tests, we'll actually run in the pipeline.

[00:16:06] Continuously, you know, uh, on every comment, uh, without even a single assertion and this really surprises me. Uh, so with LinkedIn, you can actually prevent, uh, prevent that from the able to, uh, to be checked in. Yeah. So now that was, you're already talked about that. Uh, not assertions in hooks. So let's say if you have a, uh, before each, would you say hook.

[00:16:35] So it'd be for each text, let's say I just copied it to make it faster. So let's say if an expect here, then it says what I've searched it and hugs because I'm the assertions is part of the test, right? Not part of a set up before your tasks or after a test, uh, make an assertion is a check, uh, as part of your test.

[00:16:59] Um, [00:17:00] since you to know, is that you can customize it's room. Um, so for example, uh, here, if I just copy this in the years, conflict file, you have this section called burrows and it's this one in and say, uh, off. Or I need parentheses. I always need to check this.

[00:17:35] This is the live demo effect. Yeah.

[00:17:43] This is the missing assertion in the test, right? I'm a, I'm doing the other one. I'm sorry. So I feared him missing assertion interest, so I turned it off, which is what's basically. So now you see, [00:18:00] so if I removed the expect here, I actually have a missing assertion in the test. And previously it was a red on the lines.

[00:18:08] And now, because I've turned it off, it's no longer red underlined. So if I say, okay, just do this again, then it's red on the line

[00:18:16] **Tristan Lombard:** [00:18:16] again.

[00:18:20] **Kwo Ding:** [00:18:20] So you were at the assertion hooks and no brusher commencing tests. So that's a, there was this example, do you browse a doc? Uh, so really try to, uh, uh, move it into a page objects or a different instruction layer.

[00:18:38] Be curious, you want your test to be very clean and to be readable. Um, you know, that really helps with maintainability. Don't have to a CSS page selector. So let's say, um, let me clean this up a bit to be not distracted.

[00:19:04] [00:19:00] This is still browser commands. So if you have to select, they're like a CSS selector. So cloud Swan.

[00:19:16] Yes. So basically what this rope, uh, checks is, uh, what it looks like it's, uh, tied to the page loud and it's S actually checked by, uh, how many spaces are between or. Uh, for the greater than sign. Um, if it's, uh, only one, like one space, then it's a, then it's thinks it's okay. Uh, more than that, it, it, it really seems like it's, uh, it's, it's tied to the pace layouts are really following a specific, uh, Don tree in your, uh, your source code.

[00:19:52] So that's a really bouncing to, um, flicker test or flaky locators. [00:20:00] Uh, disabled to focus tests. So if you have, um, skip it, let's skip, right. It's a disabled test. Um, People do this when, uh, when you're working on a test weight on one test is very chapel, flaky or not ready yet. Uh, we'll work on it later. You can put skip.

[00:20:21] Um, but you're sure you don't want to check the same. And if you are focused at, so let's say if have two tests like this and you just want to run only this, because I am working on the, on this test. Um, I use this a lot obviously to, uh, to, to work on one specific test to make it, uh, make it work. Uh, however, um, if you check this one in and I've seen this a lot of times happening.

[00:20:53] Uh, not your whole suite is running only this test is running. So then your, uh, build a screen. You think it's, [00:21:00] everything is great, but, uh, in reality, only one specific test is, is actually being run. So you also, you also don't want to check this, um, no, no hard weights. So this is the, the fantastic example. I won't say.

[00:21:19] Okay. That's not, um, not a very hard way. So let's say if you have a check second hard way, it actually says, well, avoid hard way, but also it's a browser command ride. You know, you will want them in the test and also, um, avoid, wait, wait in test. Uh, altogether. Uh, which is, uh, we're just this, uh, this row. Now you have either selectors specifically for what driver right now.

[00:21:52] So the link tax selector. Um, so for each role you have the different examples, um, [00:22:00] on how it is incorrect and what is a correct code for the role? Um, No, wait and touch. We talked about this and the other one is the X pass selector. So for example, here, if we have, um, no, this is not juicing experts at all because he assesses, uh, um, recommended, um, the experts selector check is actually the same as the CSS one.

[00:22:33] So if I look at this one,

[00:22:39] In here. So basically checks the, uh, the slashers. So if I have this one slash share, then it's, then it's okay. Um, like I said, you can cut so much it's, uh, to your liking to your organization. Um, obviously I'll just read a very opinionated, uh, implementation, [00:23:00] but the nice thing about yesterday is that you could customize it, uh, altogether.

[00:23:05] And also if you have your, um, um, just code is click show documentation for the role, it will actually open. Uh, well, it opens on a different screen, but it opens, uh, the specific, um, page off the, off the specific role. And just to show you, that's it. So this is a what drive Iowa example I have here, uh, Cyprus.

[00:23:34] So here, uh, actually the same. Uh, is lens conflict, but then it says UI testing Cyprus. Right. Um, and then here, there's some example tests. This is actually a, uh, example project from Cyprus itself. Um, where are you at here also, again, you absolute URL and then as you can see you see the same role, uh, applying here.

[00:23:58] Um, [00:24:00] and let's see if I have another example. Yeah,

[00:24:12] let's see. Only, just trying to have a simple example over here. Another of the focus test rules being applied, uh, the one where extra, uh, which is not related to the, uh, UI testing plugin. Uh, but a thing you can do with, with, with Yasmin. So this is the no restricted, uh, Syntex rule from your state itself.

[00:24:40] So let's open it. Here's some more description we'll know that it all works, what it actually means. Um, but the nice thing about this, uh, custom rule is that you can implement it in your project right away without. Even creating a custom plugin or installing anything. Uh, if you [00:25:00] have ears lands and you want to apply as custom rule, you can do it this way.

[00:25:03] So here I've actually the same, uh, selector or the rule. So here, if you say open quality, um, and we don't again, no one stopped this. Um, so let's say here. Mmm, open quality,

[00:25:30] right? It says this love this. And now of course, she wants to have a nicer message. So let's now this conference. And then moved back here. You see the, the, um, the RMS has been changed. So this is how you can quickly apply a custom rule in your organization if your team or multiple [00:26:00] teams, um, uh, want to follow a specific coding standard, um, or, uh, a meaningful role, you can do it this way.

[00:26:11] I'll move back to the presentation. Okay.

[00:26:19] So then there was the demo, um, some key takeaways. Um, yeah. So enable a lender for code conventions, um, and try to apply the same venture rules you already have for your application. Uh, for example, I've seen many times you have to an application. Where, um, uh, like an Airbnb code style, so already being applied to right, to apply that as well to your, uh, test code and maybe customize some rules or, uh, disabled them because they're not, um, they're not applicable for your test.

[00:26:51] Uh, but in that way you can really, uh, which is not my next point, improves the collaboration with developers because you really have the same [00:27:00] layout, same structure, and you can focus on, uh, creating awesome things together. Um, and you can who's over at the linter you can get early feedback on, on, on, on test codes and things that, uh, maybe not work, um, or really bad selectors, uh, if you already typing them in, uh, and Lynch can help you with, with that already.

[00:27:24] Um, and if you're really new to, uh linting and you're going to position, you can start small. So, uh, not just install, you know, The Airbnb coach style and, and installed the Eastland, uh, UI testing plugin and enable everything because that might be, uh, Uh, well, w it might not work that well because, uh, if everything is new and if so many strict rules, uh, your slant or, or any linter will then, uh, people won't look at it anymore.

[00:27:58] Cause yeah, it's, [00:28:00] it's all, uh, it's, it's all crap or, um, so really start small by enabling a couple of roles that you think is meaningful at the current moment. And try to expand that along the way, or, um, you can enable. Uh, rules put 'em up, put them on, uh, the, the level of warning. So it actually doesn't fill the bill, but you learn from it and then you can make it stricter along the way.

[00:28:24] Um, and the example I just gave, um, with use language, you can really easily implement a, uh, a custom rule, um, uh, It's hard to utilize that because it's, it's, it's a great, great opportunity into, uh, uh, standardize, uh, within your team or organization. Uh, and if you have more and more of these custom, uh, in rows, you can, uh, extract these custom rules into a plugin and then make it available for more people or even make it up for source.

[00:28:58] And, uh, more people can [00:29:00] benefit from it. Um, yeah, that was actually my last slide. So thank you for, uh, for a time. I hope there's some time for Q and a.

[00:29:12] **Tristan Lombard:** [00:29:12] That was an excellent talk. And, um, I was going to ask you a little bit more because, you know, obviously with ESL, like how, how would people get the, learn more about how you were able to sponsor, because you can't do this alone.

[00:29:24] Like if people are interested in sponsoring, how would they go

[00:29:27] **Kwo Ding:** [00:29:27] about that? Sparsing this project. Oh, well, so I, so I also started small, uh, as in, you know, the things that I learned in practice and putting them in into code. And I think I really believe that, um, as a community we can come up with more sophisticated rules.

[00:29:49] So, um, well I feel ready. You know, familiar with ease, Lander writing these rules, or you can obviously just create a pull request, um, if you're new to it and you [00:30:00] really want to contribute contact me and I'll, I'll get you set up or show you some examples about how to, uh, how to create all these, uh, this, uh, customize rules.

[00:30:11] **Tristan Lombard:** [00:30:11] That sounds awesome. And we put some links in below. Um, I put out the, the general that, or I put the get hub. Um, as well, we have your social. So if anyone wants to get in contact, um, Twitter, LinkedIn, all of those are there. Um, angel had a question I'm starting to, I was trigger. Happy, love this. So if we go back to the chat, you can see the

[00:30:35] **Kwo Ding:** [00:30:35] chat.

[00:30:35] Okay. So, yeah, let's see.

[00:30:41] **Tristan Lombard:** [00:30:41] I appreciate your dedication

[00:30:42] **Kwo Ding:** [00:30:42] angel.

[00:30:47] the good, uh, Oh, the examples. All right. Um, I actually have two WebDriver IO on my ghetto. Yeah. I say this, but I have to check it. [00:31:00] I think I pushed it. Um, did.

[00:31:10] Yeah, I do. So, uh, Oh, maybe I could share my screen.

[00:31:17] Here we go. Share. So here, um, So, like I said, this is basically a fork from my boiler plate project, and I upgraded it to a prepper I six. And actually this is the starting point on Mmm. How to work with it and also with the plugin installed. So, uh, if you just clown, this one, you can already, uh, play around with the rules, uh, like adjusted in your demo, uh, because it's already been, uh, been applied.

[00:31:51] Oh, Oh gosh, share this link.

[00:31:55] **Tristan Lombard:** [00:31:55] That would be great. Yeah.

[00:32:00] [00:32:00] **Kwo Ding:** [00:32:00] Here we go.

[00:32:03] **Tristan Lombard:** [00:32:03] Ask and you shall receive, and so there we go.

[00:32:07] **Kwo Ding:** [00:32:07] Alright.

[00:32:08] **Tristan Lombard:** [00:32:08] Yeah. And let us know how that works out for you, by the way, please contact whether this is a community open source project, right? So you want to make sure that we're letting, giving people feedback and I'm sure you'd like to hear that. I want to go back a little bit to you because you had a conversation with this talk back, um, at test skills.

[00:32:25] With Joe and he really, really enjoyed it. And I'm wondering, how was this project? Cause that was July, that feels like years ago.

[00:32:36] How has this project, if so, how's it evolved since then? Like what, what else has kind of changed since that last demo?

[00:32:42] **Kwo Ding:** [00:32:42] Well, actually before this plugin, I created a different plugin for the, for Java code. Uh, so for Java based, uh, slammed the code, uh, same kind of rows, uh, less sophisticated, less extensive [00:33:00] and less customizable.

[00:33:02] And I dived into the JavaScript, um, because, well, that's just getting more and more traction. Um, most of my clients. So I, I, I. Well, that's came through this and I saw many more possibilities. So it has been involved from the Java version to this version a lot. Um, and w what I love about episodes community is that I.

[00:33:27] Got some issues raised, but Hey, this is not working this case. Uh, I shared as this, this is working. Um, so I did quite some bug fixes, uh, around the test scale, uh, actually to make it more robust. This is Greg, because if you have more exposure about your projects, people are going to look at it and maybe use it.

[00:33:50] And that's how, uh, how you get more and more feedback. And, and I'd love to have more people use it. So we can really, uh, make it better and better. [00:34:00] Um, so I've been doing a lot of like buck fixes, um, and I'm thinking about a couple more, uh, rules, but, uh, yeah, I have to test them out and see how it all works.

[00:34:10] And hopefully some other people can also come up with some rules or discuss it with me, uh, happy to do that. And so, uh, because I think linting for tests, uh, We don't have many tools to do this. So I started with this. I don't think we can make it better.

[00:34:31] **Tristan Lombard:** [00:34:31] And I also appreciate you, like you obviously like asking for that feedback from everybody.

[00:34:35] And so, um, the best way, obviously you have to get hub out there so people can go, they can play around, um, and they can contact you via Twitter, LinkedIn, um, either works, right. Um, and they can share on the repo and then what we'll do as well as we'll put that information in the general chat for you all.

[00:34:53] So we can continue the conversation. And I already. Tweeted and put out on LinkedIn. So

[00:35:00] [00:35:00] **Kwo Ding:** [00:35:00] I got you. Great.

[00:35:02] **Tristan Lombard:** [00:35:02] We want to continue the conversation. Y'all this is a super important project, and I know you've put a lot of time and energy and had gotten great feedback from the community. So I want to continue to see it evolve.

[00:35:11] So the next test guilty then. More

[00:35:14] **Kwo Ding:** [00:35:14] evolved. What would you say,

[00:35:16] **Tristan Lombard:** [00:35:16] lastly, we really want, I want to, what would you say is one of the biggest takeaways? Cause you talked a little bit about how the system, something that we really have in the community. What's one, one you could convey one message from today's talk to the audience today.

[00:35:28] What would that be?

[00:35:31] **Kwo Ding:** [00:35:31] Um, well utilize all the power of the tools we have, where we're not utilizing everything we already have. If you look at applications, we're building software and you have so many cool sophisticated tools, but not for tests. And that's just crazy, because test is just as important without tests.

[00:35:50] You don't have any confidence in your software. So, and living is one part of it. And, um, like I said, yes, start small. If you're [00:36:00] really near to all these fancy tools, uh, try them out, play with it, uh, and, and expanded from there.

[00:36:09] **Tristan Lombard:** [00:36:09] Awesome. Awesome. Well said, I know it's late in your part of the world, so I just want to say thank you for, you know, Going through a whole work day and then coming here and hanging out with us as well.

[00:36:20] **Kwo Ding:** [00:36:20] Um,

[00:36:21] **Tristan Lombard:** [00:36:21] so I'll put the information in the chat, but, um, w if you're around a little bit for minute, I'm not sure if you're just signing off right now. I can put that in the general chat.

[00:36:31] **Kwo Ding:** [00:36:31] I'll be, I'll be here for awhile. Okay. Cool.

[00:36:33] **Tristan Lombard:** [00:36:33] Awesome. Awesome.