

Ep 36 Trevor Lohrbeer

Francis Wade: [00:00:00] The Task Management and Time Blocking Podcast, Episode 36 . You're someone who does a great deal of time blocking in your calendar. You follow the best practices and don't try to schedule big complex tasks. Instead, you schedule their subtasks. But how do you deal with tasks whose complexity is hidden? In other words, the task that you are engaged in contains far more subtasks than you imagine.

Francis Wade: And it's a bit like watching a four hour task turn into a 20 hour task. When you're on the receiving end, it's not pretty. It can be very upsetting, but is there anything you can do tune into this episode to hear from me and my special guest, Trevor Lorbeer as we tackle this wicked productivity problem together.

Francis Wade: I'm Francis Wade and welcome to the task [00:01:00] management and time blocking podcast. And we're back. And as you can see, I'm joined on the stage by my special guest, Trevor Lorbeer. This is Trevor's third time with us here on the podcast. And there's a reason that he's been invited back this many times, more than anybody else.

Francis Wade: And in a minute, I'll introduce him. But if you're new to the show, you may wondering what exactly is going on here. This doesn't look like the traditional podcast and guess what? It's not. So this is not an interview. As bright and as compelling as Trevor is, he's not here to be interviewed. He's here as a guest to tackle the wicked problem that I mentioned at the beginning of the podcast.

Francis Wade: And what you'll be listening to is our attempts to diagnose the problem. We're trying to share with you some of the underlying thinking, trying to pop the hood. Put up the bonnet so that you can actually see the inner workings of the kind of things that we do when we're [00:02:00] trying to solve these problems.

Francis Wade: And we may present the final answer to you at some point, if we arrive at one, but it could be more interesting to hear exactly how we got to it. So here in this particular case, we don't have a final answer to this particular problem. But we're going to share with you our best thinking or best intuitions and our best insights.

Francis Wade: And hopefully by doing so, you'll be able to take them, apply them to your own life and use them in your own situation. We're following what Einstein said. He said, if he had an hour, he'd spend 55 minutes thinking about the problem and five minutes thinking about the solutions. So we're going to be diagnosing and we'll also have an insight bell, which I'll ring when one of us comes up with an insight that we're going to The other person has never heard before.

Francis Wade: Why? Well, because coming together is more than just sharing ideas. We want a little bit of synergy to happen, some serendipity, so that we put our best ideas together and come up with something brand new that [00:03:00] maybe moves the needle for the first time ever. So to help us on our way, let me tell you a quick story, then I'll introduce Trevor.

Francis Wade: Evelyn, she's due for a promotion to become a manager. And her leadership team is watching to see if she has what it takes to function at kind of the next level. She's nervous as this is not a guarantee and she needs to support her son's acceptance at a very expensive college or university. So she has some bills.

Francis Wade: that are coming due and she needs the extra income. Recently, however, she missed a major deadline by almost two weeks and now she realizes that she must up her games, so to speak, improve her performance. And a little bit more background, for the last year she has been assigned to these unique projects, kind of first time, one of a kind [00:04:00] sort of efforts.

Francis Wade: They're the most difficult to plan and predict. because of the number of complex tasks involved and that's why she missed the last deadline and is missing more than a few here and there. A seemingly simple task exploded and became far more difficult all of a sudden. So she's got to find a way to cope, but she's really not sure how to proceed.

Francis Wade: And that's why I invited Trevor. So Trevor Lorbeer is the founder of Day Optimize, a digital day planning web app, and he's also a productivity expert. Since the release of ChatGPT, he's turned his passion toward AI, learning how to use, configure, and build AI chat applications to increase productivity and develop new ways of interacting with.

Francis Wade: Ai in november 2023 a few months ago He started the ai meets productivity podcast with a gpt. He customized [00:05:00] to be a podcast host Trevor welcome to the show. And what do you think about evelyn? Thank

Trevor Lohrbeer: you so much. It's good to be here. Um, I think evelyn has challenges that we all Uh face from time to time as you know, we all run into these hidden tasks that um derail us and make Uh our deadlines Seem like phantom deadlines, not, not real.

Trevor Lohrbeer: They just kind of have to keep being pushed back because we keep running into things that we didn't anticipate.

Francis Wade: And when we're on the receiving end of these, I don't know about you, but my experience is one of heightened frustration. Um, and I start desperately looking for shortcuts. And it doesn't work to find these shortcuts, because when the task is truly complex, it means that whatever plans I had for the rest of the day, uh, or week, depending on how bad it is, they go out the [00:06:00] window.

Francis Wade: So, you know, we all want to be, if you're listening to this podcast, folks, chances are you want to be super productive. And then there are these moments. We run into these kind of hidden bombs, and so not only does the task take longer, but there are other problems too, right Trevor?

Trevor Lohrbeer: Oh yeah, I mean, I think one of the things with them that affects your productivity is this, when there are unexpected hidden tasks, you, We often find ourselves getting way more stressed out about them and they kind of derail our productivity because suddenly you were like, we're going along, we thought like I had an hour for this task and a bunch of other hidden tasks popped up and not only like, is that task itself now.

Trevor Lohrbeer: Delayed and taking longer, but you kind of feel off your game for the rest of the day or the rest of the week of like, wait a second, like it's, it's disorienting when you don't know [00:07:00] those hidden tasks are there. So I think one problem is like, there's going to be tasks that we, we don't know what steps to take.

Trevor Lohrbeer: Are going to need to happen to actually finish the task. Those are the classic hidden tasks. But the second, second degree part of that problem is that we're not even expecting hidden tasks to be there. And that

surprise can affect our productivity because it can derail us and disorient us. Right.

Francis Wade: Right.

Francis Wade: It means that the rest of our week and whatever buffers we had built into or, or nice, nifty, optimized schedule, go to heck. And now we're having to, like, you know, like Evelyn, we're having to explain to our uppity ups, people who we're accountable to, that we're not going to be reaching our targets. So we're, you know, some uncomfortable conversations now ensue.

Francis Wade: Um, And, you know, the kind of things that can keep you up at night. And it's all [00:08:00] because you came upon this hidden task that you didn't, and you just didn't know. And we'll talk about this, but you know, when you're in the middle of it, you can feel very anxious and stressed as Trevor said, and, but, but in terms of, you know, in terms of sort of the bigger picture, this is not a problem that will ever go away.

Francis Wade: This is a problem to be managed. It's always been there. It has always been in anyone's. thinking, you know, it should be pretty easy to do this part of the task or the job of the project and you allocate, mentally allocate a certain amount of time to it and then you, you get a rude surprise. But this has not been, this is not new and they're building the pyramids.

Francis Wade: Right? They, I don't know what problems for the purpose, but I'm sure they had some of them, right? The sand is not the proper sand or the foundation is going to take [00:09:00] 10 times to build or whatever it is. There's going to be, there's always something. So question is given the epic nature of this problem and the fact that it's always around, it's never going to go away.

Francis Wade: What can be done about it? Are there intelligent ways to approach it so that you can. Mitigate its effect. You can anticipate. when it's going to happen. You can respond more effectively in the minute, in the moment. Um, any other outcomes you can think of Trevor?

Trevor Lohrbeer: Uh, no, I think those are good. I think, I think to me the two things, uh, the problem breaks down into uncovering them as much as we can and then anticipating where they might be when we can't uncover them.

Francis Wade: Right, right. And there's, there's, you know, there's probably things that, you know, I know if I'm, if I'm, if I'm on Evelyn, I, I don't want. People to think I'm going to be incompetent. [00:10:00] So I'm not necessarily going to, you know, I don't want to telegraph that I don't know what I'm doing. And I don't want to jeopardize my promotion.

Francis Wade: Um, I want, I want to be effective and I want to appear to be effective. So there's also some ego involved, um, for me. Trevor, you might be a different kind of, uh, more evolved individual, but for me, you know, I want to look good and have them like me because I, I hit my deliverables.

Trevor Lohrbeer: Well, I think it's, it's not just, I think you can have the ego aspect of there, but it's also the effectiveness of you is like, if people view you as incompetent, they are not going to interact with you in a way that helps you be competent.

Trevor Lohrbeer: And so there's almost, you need people to view you as being competent to work with them effectively to be competent. So even if you're totally competent, but everyone views you as incompetent, then they're maybe not going to respond to you as quickly. They're not going to treat [00:11:00] your, what you say is, um, worth the same as what someone else says.

Trevor Lohrbeer: And then that itself is going to make you incompetent. So yes, it can be an ego thing, but I think it can also be an effectiveness thing is you do need to be perceived as being competent. Competent to be competent.

Francis Wade: I agree. And I imagine that Evelyn is kind of maybe You know, she wants to, she wants the promotion, so there's a perception involved, no matter what she does, and she realizes that this promotion could be at risk, right, because she's not delivered, and as she goes along, as she looks to the future and the kind of work that she's doing, she may be thinking, why am, why do I think I'm going to be safe, and, and probably when she was doing previous kinds of more repetitive work, she probably had a sense of confidence because she was doing the, you know, You know, the tasks, these compositions, she was breaking down tasks into subtasks, and she was highly effective, you know, [00:12:00] because she was in a more predictable universe.

Francis Wade: Now things have changed and she's on these different kinds of projects, and her confidence turns out to have been misplaced, or maybe it was hubris, or she thought she had handled the problem. But it's a good training ground because if she gets to be a manager, you know, hello, she's going to have people who are working to her, working for her, who are going to give her all kinds of false confidence that they know what they're doing when in fact they don't.

Francis Wade: Right.

Trevor Lohrbeer: Yeah. And I think the one thing we talked about, like the skill that she'd learned in the projects that were. Very predictable that idea that I'm going to take my project, break it down into tests and subtests, and be very detailed or classic project management skills that we learn. It's like, Hey, if you, if you've got a task break into subtests, so you know what needs to get done.

Trevor Lohrbeer: That can also be a source of the problem in that. By doing so you gain this false sense of confidence that you then have [00:13:00] completely described the problem because I've broken it down and I've thought about ahead of time. So therefore I should have captured everything and maybe I even went and allocated time and estimated time for every subtask.

Trevor Lohrbeer: And so like, I'm very confident with my, um, time estimation for the whole project, but. If we don't think about the hidden tasks in there, that's where it can derail us. And then that's what, again, causes more stress because we had this false sense of confidence. So having a little bit of humility there going, yes, it's important to break down everything into subtasks and, and kind of identify everything, but also recognizing you are probably missing something.

Trevor Lohrbeer: And so how do we find out what you're missing and anticipate either uncover what you're missing or anticipate that, yeah, this is an area where I just know I'm going to be missing something. So be prepared for those things to pop up that you didn't think about.

Francis Wade: Right. So, so, and I [00:14:00] think that those of us who've been time blocking for a while can feel that sense of overconfidence because it works so well when tasks are understood, right?

Francis Wade: It does are better than the other guy who doesn't time block.

Trevor Lohrbeer: And it goes back to every, every technique, every piece of advice is context sensitive. So like, if something is predictable, then time blocking works really well. Breaking down your task works really well. Once you go into these areas of unpredictability, then that strategy and those techniques break down.

Trevor Lohrbeer: It's not that completely useless. You still might want to use them, but they're not going to completely solve it. So like, if you, I know you write an article, I forget if it's every week or every month, but like writing an article, you probably have it down. You know what your subtasks are. There's very little, very few hidden tasks in that.

Trevor Lohrbeer: Cause you, you just repeat it over and over again. You can [00:15:00] time block it accurately. That's very predictable. Whereas if you are saying, okay, I want to go and create, um, a new AI chatbot with all the content I created. Well, you've probably never done that before. And so there's going to be a huge number of unknown tasks in there.

Trevor Lohrbeer: These hidden tasks that are going to completely derail you.

Francis Wade: I didn't even know I could do that. So you're going to tell me after our call, how I can do that. You can do that. This one is used to be okay. So we're going to have a conversation. I'd like, you know, I'd like a nice shortcut when it comes to these things.

Francis Wade: And I want to reduce the chance that, you know, I'm going to run into one of these unexpected, unwanted kind of complex tasks that will just end up derailing me, derailing me. Um, before we came on the call, we talked about the difference between an execution and exploration planets. It's [00:16:00] here in my notes and maybe Understanding the difference between the two could give a different kind of confidence.

Francis Wade: I know this is kind of solutions, but do you recall the distinction?

Trevor Lohrbeer: I'm trying to remember now. Let's see. Uh, what were the two things? I thought it was your

Francis Wade: brilliant insight. So that's what I gave it to you. It's like, we're talking about here.

Trevor Lohrbeer: Um, Oh, actually, I see exploration. I misheard, uh, execution versus an exploration plan. Yes. Um, so the, the idea here is that distinction between, um, I've got a list of tasks or subtests that I'm just going to ask you to go straight through those versus the idea that, hey, I want to explore because there's going to be, you know, hidden tasks within here.

Trevor Lohrbeer: So how can I take those things and explore to kind of surface as many hidden tasks as I can before [00:17:00] executing? And so a lot of times we just think about what is our execution plan? What is, how do we break those things down? The exploration plan is going down and looking at each of those tasks and trying to identify, Okay, which of these tasks might have hidden tasks within them and then exploring that up front to uncover them more.

Francis Wade: Right, let's expand on that when we get into solution mode because this points to, I think it's the first insight of our time together or insight bell. Um, bring that, but this is the first, this is the, uh, a key differentiator because time blocking is typically thought of as a bit mechanical. Tell me if I'm wrong, you know, as, as, which is where the false confidence comes in.

Francis Wade: Because when you're mechanically time blocking simple stuff, you, [00:18:00] you do get on a roll. You know, you do, you're, you're able to just kind of crank out the widgets. But then life intervenes and along comes a complex task and your old way of cranking out the widgets doesn't work. Um, I think it's one of the criticisms that people have of time blocking and why they say you shouldn't, some people say you shouldn't do it because you can't be sure that you're going to be able to follow your schedule.

Francis Wade: So don't even bother to try. So that's a little bit of throwing them, maybe out with the bathwater, right? It's like, oh, you might die one day, so don't, don't even live by that argument, but, but, um, let's, let's, let's, let's hold that and come back to it, hold that, that point and come back to it just a little bit.

Francis Wade: Um, so th this business of overconfidence is, is kind of what we're, we're imagining that Evelyn is experiencing, she's experienced in the

past and why she's so thrown at the moment. Um, but if she accepts the fact that this is, [00:19:00] this is what life looks like now and what it's going to look like in the future, there's going to be more of it, not less.

Francis Wade: I think she would have a very different mindset that maybe that mindset is shows up, shows itself in the exploration versus execution kind of distinction. So we'll come back to that too. Um, unfortunately, if she, or actually listen, let me, let me talk to you. You may be listening, trying to gain some facility or gain some skills in dealing with all kinds of, all kinds of tasks and all kinds of challenges, especially those that are kind of semi creative and You want to get to the point where you're not blindsided and thrown off course.

Francis Wade: Um, especially if you are more of a mechanical time blocker that these things can not only throw you off course, but they make you feel like you're not productive anymore. Some people would say, well, I'm going to stop time [00:20:00] blocking because clearly it doesn't work. They may even go to that extreme. Um, but you may want to leave.

Francis Wade: Here with more of a, I wouldn't say a buffer, but with more capacity to be flexible, um, because what we're hinting at is that this is a part of your life, as long as you get past a particular level in a company, and particularly if you get to the management level, this is going to become a problem, more of a problem.

Francis Wade: Would you say that's accurate, Trev, or do you see it differently?

Trevor Lohrbeer: Um, no, I do see it differently. I think it, I think, um, at the manager level, things can, to some degree, become a little bit more predictable because a lot of times, like your, your life just becomes meetings, meeting, meeting, meeting, meeting, so the overall, it's the nature of the problem changes because at that point in time, your unknowns are managing other people who are [00:21:00] Are encountering the it's unknowns versus yourself.

Trevor Lohrbeer: So it really kind of depends upon like what the, the work that you're managing is doing. So again, if you are developing a new product, you're trying to like invent a new, a new drug, a new technique, a new, whatever invention is going to have a lot more unknowns than say, If I'm running an

editing company and all we do is we get in manuscripts and we edit them, there's going to be a lot less unknowns as a manager in that case.

Trevor Lohrbeer: So it's really what the work is doing rather than if you're a manager or, um, an actual employee doing that work.

Francis Wade: Okay, interesting. My, my, my, I'll share with you my reasoning, um, is that you have people working for you as a manager and they're, they're the source of variability, but they could be reliable workers.

Francis Wade: Um, my general assumption is that. [00:22:00] When you, when you get people working for you for the first time, you're blown away by the, you try to compensate by doing their work for them. But the first thing that hits you is that they don't know what they're

Trevor Lohrbeer: doing.

Francis Wade: And this is an interesting

Trevor Lohrbeer: point. It's like being able to make this distinction between predictable and unpredictable work in evaluating your employees.

Trevor Lohrbeer: Right? Did I give my employee a predictable task that was at their level, their skill level, so they should have been able to do it in a predictable time, or did I give them an unpredictable task, and I thought it was predictable, and now I'm unfairly judging them because they weren't able to meet that deadline.

Trevor Lohrbeer: But there were a lot of hidden Unknowns within that task that they had no ability to, to evaluate. Maybe I could have actually evaluated it because I have more experience in this. I could tell, I've told them where the hidden unknowns were, but that idea of. [00:23:00] Yeah. Figuring out how to evaluate the incompetent employee from the employee who is just struggling with a new task they've never done before that has a lot of hidden unknowns.

Francis Wade: So,

Trevor Lohrbeer: you know, if I, if I hire a social media manager who's supposed to be managing my social media, I'm going to expect them to be able to post on Twitter, post on Facebook, you know, do all that kind of thing. There

should be very few unknowns in there. So if they're not meeting the deadline, I can say, okay, that's probably due to their skill level, their competence level, et cetera.

Trevor Lohrbeer: But again, if I hire, um, you know, I'm a big pharma and I'm hiring someone to do drug development. And I say, okay, you need to like create this breakthrough drug in three months and they don't create in three months. Is it my fault for setting the three month deadline or is it their fault? Because they weren't able to like invent something new in three months.

Trevor Lohrbeer: Um, again, different, different scenarios, you know, how, [00:24:00] how likely is the work to have these hidden tasks? And that then helps us evaluate. Our employees and helps us learn how to manage that as well, because, you know, part of our job as a manager is going to be to surface those hidden tasks that the employees may not see.

Trevor Lohrbeer: So there's they're always going to be for some projects. There's always going to be hidden tests that no one knows But there's some hidden tests that I know that my employee doesn't so how can I communicate that to them and say? Here's the hidden tests you're going to need to account for and deal with in this process.

Francis Wade: Yeah. Yeah, I I so Let's let's move into solution mode because I i'm i'm thinking I actually had this situation. Um, so I have a I have a You And kind of an intern, uh, who works, who works for me. And she took over a task that someone else used to do for me. So I had one intern replace another [00:25:00] intern and a prior intern.

Francis Wade: So he went through, you know, he had a, they were all unknowns to him at that point, and some of them were unknown to me because it's one thing to be able to do something yourself and then give it to somebody else to do if you've been doing it for a while, because you've forgotten all of the complexities that are involved.

Francis Wade: Because you've already figured it out, right? So when I handed it over to him, he was, you know, for some things he was like, what? And I repeat the instruction and he'd be, what? It's not that he's dumb, smart guy, but it's that I have things that I understand about the job that he doesn't. And if I haven't prepared him properly, if I hadn't put, I didn't put myself in his shoes and say, what's it like for him receiving this particular instruction or.

Francis Wade: Receiving this task or project. And I wouldn't see the complexity and I wouldn't anticipate that he, you know, he [00:26:00] says, you know, I took me a whole five hours to enter two names and I'm like, what was it take 10 seconds. And then he explains that I'm like, Oh, I totally forgot. Now that I have another, his replacement, I'm way more careful, uh, to explain things to her in a way that didn't explain them to him.

Francis Wade: So she's facing far less, less complexity than he did, but that's only because I went through, you know, I made all the mistakes the first time around and second time I've kind of figured it out, it makes me think that there's, there's something, I wonder if, I don't know if it's extraordinary. But there's some way in which I need to, when we're delegating, we need to put ourselves in the shoes of the other person so that the complexity is reduced.

Francis Wade: And I can't tell you that that's easy because having just made a mess of it, I think it's actually really hard, but [00:27:00] it makes all the difference in the world to the person who's going to do the task. What do you think?

Trevor Lohrbeer: No, I totally agree with that. Like, I mean, understanding the context within what someone's operating in their, their life experience, their skills, their knowledge.

Trevor Lohrbeer: In general and about the work you're doing it's critical to give them the right information To be able to do it. Um, I think one of the the struggles sometimes the the mistakes people make for instance when also outsourcing Overseas or right it can even be to any any different culture. Is that the cultural assumptions are different

Francis Wade: for example

Trevor Lohrbeer: Yeah, yeah, and and and we have we have certain assumptions that we don't communicate Because there are cultural hidden, hidden assumptions in our culture.

Trevor Lohrbeer: And then when we interact with someone from a different culture, if we don't surface those assumptions, they're going to bring bringing their own assumptions. And then, [00:28:00] so how they interpret what you say is different. But the same can apply to like people working in, uh, different types of companies have different types of cultures.

Trevor Lohrbeer: And so like, if I came from culture X and I'm going to culture Y, I might need to make that translation, that, that adaptation as well. Um, so yeah, hidden assumptions are similar to hidden tasks. Like what are the things that like, we're not, sometimes we're not even aware of. So we, we don't know how to communicate them, but certainly, um, when we are aware of them.

Trevor Lohrbeer: Or when we can at least introspect to become aware of them, how do then we communicate them?

Francis Wade: Right, and I also noticed in the, you know, the fellow I had working for me, he became faster at certain tasks than I could be. So he, he, by virtue of repetition and also, you know, he's, he's younger and he's a bit more snappy with his thinking than I was, he figured out ways to do things faster [00:29:00] than I could.

Francis Wade: He, so if I asked him to do something and do something with a hundred names, you know, 15 minutes later, he's done, you know, I, I couldn't do it that fast, but so there's also accounting for, um, a learning curve. Okay. When it comes to delegation, no, and even with ourselves, I think there's a, uh, a learning curve.

Francis Wade: It's just that. We're more likely to think that we're higher up on the learning curve than we are when it comes to dealing with complex tasks. And if that's a built in thing, what do you think?

Trevor Lohrbeer: Um, I don't know if we're, we think of ourselves as higher up, but again, I think it's very easy to, um. Forget about those hidden tasks. Which are the things that, that derail us a lot of times in that. So it's when you have a complex test, we make the mistake [00:30:00] that it's a simple scale thing that like a complex task is just a bunch of simple tests put together, and that's not true.

Trevor Lohrbeer: So it's just about

Francis Wade: that.

Trevor Lohrbeer: Um, so going back to going back to the, um, test breakdown that most people do. It's like, okay, I've got this complex task. Let me break that down into a bunch of subtasks. Let me simplify that into simpler and simpler subtasks, and then therefore I'll be able to do. The entire task more

well with a complex task, even if you could identify all the subtasks and right now this in this episode, we're talking a little bit about hidden tasks.

Trevor Lohrbeer: We'll get to that in a second. You have dependencies between those tasks that you don't with the symbol test. So just that adding dependencies between simple tasks increases the complexity. But then we have these hidden tasks. Which also have dependencies with those tasks that we've identified. [00:31:00] So not only with a complex task, do we have a collection of simple tasks that are dependent on one another and affecting each other, we also have these tasks that we don't even know about hidden embedded within that complex task.

Trevor Lohrbeer: So those are two issues why a complex task is not simply. Uh, scaled up simple task.

Francis Wade: Programming is, involves a lot of complex tasks, which with, with unique, unforeseen dependencies. People who design apps tell me all the time that, not all the time, whenever I meet them, that um, you think an app is so easy.

Francis Wade: You know, because, you know, I'm the kind of guy who says, why don't you just develop an app for that? And they're, and they look at me like, you idiot, you think it's that easy? Developing an app is way more, you think it's easy and you know, the concept seems straightforward and [00:32:00] you actually try and go to do it and you, you won't believe the complexity is what they tell me because the interrelationships that they need to master or they need to translate into code.

Francis Wade: Are just at a whole nother level of complexity than you think of when you think of oh Why don't we just put an app together for that and two aren't two different different worlds, but you're a software guy You tell me

Trevor Lohrbeer: Well, let's dive into that because I think that this is going to be a useful metaphor for a lot of different particularly client based projects, but it could also be for your boss too is say, um, you want to Like I get hired to do app development, right?

Trevor Lohrbeer: One of the things or anyone gets hired to do app development or any project One of the things you often want to do with your client is to show progress early on because like they hired you Um, and so what

you tend to do is you tend to do The surface tasks the easier tasks So maybe like what I do is like, Oh, I let me design [00:33:00] model out the whole interface, you know, and save the complexity in the back end that no one's going to see till later because you want to be able to show progress to your boss, your client.

Trevor Lohrbeer: The problem with that approach is now what we've done is we've taken the tasks that are more likely to have hidden unknowns and pushed them toward the end of the project and done all the easy tasks up front. And now we've basically created our own time bomb. We're like, you know, we get this, the client now thinks we made so much more progress and we're, we're working so fast.

Trevor Lohrbeer: So they're all excited about the deadline we created. Um, or your boss is super excited. And then when you can't meet the deadline, it's like, well, you got that, you got everything Gave me that beautiful interface or that beautiful, like first draft or whatever in the first month and it's three months later, how can you not have it done?

Trevor Lohrbeer: Instead, what we need to do is we [00:34:00] need to put our highest risk tasks first. And do those the ones that are most likely to have hidden unknowns first go through that exploratory thing that exploration plan of where we need to identify and try to even if we don't finish those tasks, explore them enough to uncover as many hidden unknowns as we can.

Trevor Lohrbeer: So that way we're doing. The high risks of first, and then that also then changes the expectation. If, if I say, Hey, it's going to take me two months, three months before I can show you an interface, then, then when I get my deadline, like a month later, two months later, the person's going to be so excited.

Trevor Lohrbeer: Wow. You were able to like finish that up really quickly. It took you a long time to get started. And if I'm not experienced at working with this type of person, I might be like disappointed, like, well, this other guy here could easily get it done much faster. Now, that's not true. What they could do is they could show you [00:35:00] rough draft much faster.

Trevor Lohrbeer: But they can't finish it much faster. You know, it's that classic, uh, the 90, 90 problem, like the 90 percent of the test takes the first 90 percent of the time. The remainder, 10 percent of the test takes the other 90 percent

of the time. And so, so often, and that's because what we do is we do, we, we frontload all the easy tests, all the tests that we're really certain about that we're comfortable about that help show progress to our boss, to our client about.

Trevor Lohrbeer: In the front and all the tests that have like huge unknowns that are behind the scenes or under the water You use whatever metaphor you want all those tests. We push toward the back of the project And those are the ones that then take the other 90 percent of the time. So what I would say is one strategy there is to like pull all those tasks, go through the process of trying to identify, you know, [00:36:00] categorize, this is a type of task.

Trevor Lohrbeer: Like it's a invention task, an architecture task. It's something that is going to have a bunch of hidden unknowns. Push that to the front of the project and all the easy things like deciding what color to use or what, whatever you're going to do. Like, you know, if you're putting a report of like, okay, formatting the report, put all those toward the end.

Trevor Lohrbeer: When you know that all that is predictable.

Francis Wade: Oh, well, that's actually, it's, it's, that's a two, two insights, two insights in one. So the first one, I didn't ring one for complexity. The idea of complexity versus hidden. So complexity meaning interdependence. And then the other is the tendency, the psychological tendency for us to do the non complex.

Francis Wade: Less interdependent, more understood, and more known activities upfront because, you know, you, you offer the low hanging fruit supposedly, you know, , I low hanging fruit is a, is a [00:37:00] nice idea when you don't have a lot of interdependency. But I guess what you're seeing is low hanging fruit is a problem when what you're really doing is delaying the interdependent, delaying the hidden, hidden interdependencies.

Francis Wade: Until you have 10 percent of the time left and that's where you're, you're basically shifting the risk to the wrong place and you're creating problems for yourself.

Trevor Lohrbeer: Let's use the low hanging fruit metaphor, right? It comes from picking fruit off a tree. So if I go and say like, I need to pick all the food off

this tree and I start with picking all the low hanging fruit and spend like, you know, five hours picking the low hanging fruit.

Trevor Lohrbeer: And then I'm like, okay, well the tree is half done. So it's only going to take another five hours to do the rest of the tree. No, those are, that's the fruit. That's all the way high up in the tree that I need to climb. I need to like go out on that little branch that might break, you know, Maybe it does break.

Trevor Lohrbeer: And now like, you know, I need to like heal my broken leg [00:38:00] while I'm trying to like, pick the rest of the fruit, or I need to go get a ladder now. And I didn't think cause the low hanging fruit, I didn't need a ladder, but the other food I need a ladder for, you know, all the complexity there, you know, if instead we started with the stuff at the top and at the end, you just pick the low hanging fruit, the low hanging fruit is easy.

Trevor Lohrbeer: We can predict it. There's, you know, so then we are able to meet our deadlines. Much better.

Francis Wade: Also the, the, the, the, the low hanging fruit it because it's, it's so near to you. And so there's some trees that we have here in the Caribbean, for example, that when they're laden with fruit, the fruit hangs low and as you pick fruit.

Francis Wade: All the fruit rise because the tree doesn't have as much weight to carry. So you've, you've made it even doubly, doubly worse by going with that strategy. You know, psychologically, it feels good to do that, but practically [00:39:00] in terms of getting the entire project done, if it involves all the fruit, then it makes no sense to pick the low hanging fruit.

Francis Wade: First, you're better off, as you say, going for the complexity, because there actually is an interdependence on some trees between the fruit. The order in which you pick has a big influence on the total time it takes to pick all the fruit. We might be straining the metaphor here, but

Trevor Lohrbeer: No, no, I think that's useful.

Trevor Lohrbeer: Are you following this? I think that's useful because if you think about a lot of times, the other factor that comes into play is if I do like the easy tasks first, And then I just go to the hard tasks that uncover a bunch of

unknowns. Sometimes those unknowns change the nature of those early tasks. I have to go back and redo them or I didn't need to do them at all.

Trevor Lohrbeer: So like I'm winding, doing like unneeded work in the beginning. Whereas if I did those tests with the high unknowns, I'd be [00:40:00] much more certain. That the work I do after that is going to be appropriate to the project and I'm not going to need to redo it. So again, like, yeah, by that, that's similar to your metaphor.

Trevor Lohrbeer: If I tackle all the low hanging fruit, it changes the nature of the job, everything else that comes after it. And, and in the task world, it means like, you know, I decided to develop the whole interface. Before I dove deep into the complexity of the problem. And then once I go into the complexity of the problem, I realized, Oh, wait a second, this problem has these certain characteristics.

Trevor Lohrbeer: We need a completely different interface. So I went and built everything based upon the easy stuff, the low hanging fruit, and then realized I have to go back and redo it. And so that introduces even additional work, additional time and delay in your project, because you're redoing stuff that you wouldn't need to have done.

Trevor Lohrbeer: How'd you start it with. the more difficult [00:41:00] tasks first, the higher risk tasks first, and de risk your project, and then going forward from there.

Francis Wade: So the higher risk, the more hidden, the more complex, the ones you have the least information about, or the least knowledge about. Just repeating that for those who are listening, that, you know, for Evelyn, it's go for those that introduce the most risk first.

Francis Wade: Because they'll reduce the risk the most. Yes.

Trevor Lohrbeer: Correct. Yeah. Yeah. So there's this concept, uh, and we talked a little bit in, um, episode 10, the, how can you fix your time at optimism podcast on, uh, your risk curve and how do you reduce your risk curve? And the idea is if I love that, you remember what we talked

Francis Wade: about.

Trevor Lohrbeer: Six months

Francis Wade: ago, last week, I track all of my own. So [00:42:00] let alone what we did, please break it down for us as if I wasn't there either. Go ahead. It's

Trevor Lohrbeer: having a note system. I don't have my memory is shit. Um, the, the idea there is if I take all the, um, Tests on project and I assign them a risk score. Um, so let's say I assign them a risk score one to five.

Trevor Lohrbeer: And then I add up all the, the scores. Um, then what I have is the total risk for my project. So maybe for one project, I have a risk of 200. And like, say it's a, a 10 month project, right? So a linear way to reduce your risk scores at 200, I'm going to just chop away at test with 200, uh, 20. Risk points per month, and then I'm just kind of going down.

Trevor Lohrbeer: Um, well, usually what's gonna happen is you're gonna have a huge number of, of tests that are gonna [00:43:00] have like a risk score of one or 10, uh, um, two or something. Uh, let, let's actually use one to 10 as the risk thing. Um, and then I'm have other tests. Lowest risk score. One is the highest risk score, one is the lowest, and 10 is the highest.

Trevor Lohrbeer: Okay, so like, sorry. And you can use different scales, like, I mean, you might wanna use one to 50, one to a hundred on the risk. Right. Um, okay. And the thing is, if I, if I, so say I've got a task that I tackle that's has a risk score of one. Well, I just reduced it by one out 200, now I'm the 1 99. Mm-hmm . If instead I do my task that has a risk score of 10.

Trevor Lohrbeer: Well, now I'm not 1 90. You know, I do another 10 tasks. I'm down to 1 80, you know, so you can see that I can greatly reduce my risk. Very early in the project by doing a couple of those key high risk tasks versus doing a whole bunch I'd have to do on that sort of scale 10 low risk tasks just to meet [00:44:00] one high risk test But that's what we tend to do because they're easy and because we know what to do And and because they show progress to our boss our client again, though What we want to do is we want to take that risk and rather than it just being like Staying super, super, super high, super high.

Trevor Lohrbeer: And at the very end, we're going to try to do all the high risk things and bring us down to zero. So like 10 months, 10 month project, it's

month eight and we start at 200 and we're now at like, you know, one 50. And we think we're going to try to do one 50 points of risk in two months. not going to happen.

Trevor Lohrbeer: Um, or there's the linear, okay, I'm going to try to do 20 points a month. And if you can sustain that, great. Um, what is more effective is if I can try to bring all the risks tasks, all the high risk tests at the beginning. So it has this kind of reverse S curve. Look to it. So where I start with 200 in the first month and I try to go to 150 100 and now like, you [00:45:00] know, and now like at the very last couple months.

Trevor Lohrbeer: Yeah, then I'm then I'm like 80 60 And bringing it down slowly after that, but you try to rapidly reduce your risk up front and then handle like the lower risk stuff

Francis Wade: Interesting trouble for that data.

Francis Wade: It, um, it's more an internal resistance, you know, than a problem for the recording. Um, um, all right, I'm going to pick up where I left off. Three, two, so interesting because now you're painting a picture for me around project, project management or task management as a risk consuming activity rather than a time consuming activity.

Francis Wade: Because usually when we think of time blocking, we're thinking of, I'm sure Evelyn thinks of this, you know, that, [00:46:00] okay, well, I have all these things to do and they're all, they all have an assigned time and the chat, the choice of what I do first and second is probably more about convenience. than it is risk.

Francis Wade: I imagine that most time blocking folks, as you're listening to this podcast, listeners, you may not think in terms of risk, but what Trevor is saying, essentially, I think, is that if you're dealing with complex tasks, you must think in terms of risk. When she was a, you know, when Evelyn was, you know, five years ago, when she had simple, Stuff to do easier tasks with more, more regular or more predictable.

Francis Wade: That wasn't as much of an issue. Now that she's moving into more complex projects and she's time blocking her time, no, she needs to bring in risk as a method of prioritizing what goes in our calendar and what, and

[00:47:00] when. What do you think Trevor? Am I accurately representing kind of where you're heading with this?

Trevor Lohrbeer: Yeah, I mean, I think that's accurate. Like, I, I think to the degree possible, and it's not always possible, doing the higher risk items first are going to help you meet your deadlines more than, uh, The typical strategy, like I said, which is to either think of the project as a linear project, and I just do step one, two, three, four, or a lot of times what people will do is I'll just grab the easy task to gain momentum, but then they just keep grabbing easy tasks and they don't switch to the hard tasks fast enough and they defer them to the end because they're uncomfortable because they have a lot of unknowns.

Trevor Lohrbeer: And so those are the recipes for disaster.

Francis Wade: I've never seen time blocking portrayed as a risk consuming activity. I've never, never seen that. I've seen, you know, as I said, time consumption or effort consumption or energy consumption. You know, you spend [00:48:00] high energy in the morning and, you know, we, we, we talk about it in terms of those, you know, our deadlines.

Francis Wade: You know, which, how, how, how, you know, when will the task be due and you work on the ones that are the most urgent, but not in terms of, of, of risk. And really risk is the right, um, filter to use if you're going to be talking about hidden and complex tasks. No.

Trevor Lohrbeer: And let's, let's real quick connect, um, risk to time for people.

Trevor Lohrbeer: So a, you can measure risk in different ways. One way to measure risk, um, is to say, what is my time variance in doing the task? Right. So, uh, a task with a risk level of one means if I. [00:49:00] You know, estimate it's going to take me an hour. Pretty much takes me an hour. You know, if you look at best case, worst case, best cases, an hour, worst case, an hour expected case, it's an hour.

Trevor Lohrbeer: So what we want to do is connect time to risk.

Trevor Lohrbeer: And so if you think about a, Um, one hour task. If it has a risk of one, then that means that, like, It's going to take me an hour to do, which basically like if we say best case, worst case, best case is an hour, worst case is

an hour. It is, that's a risk level of one. If we think a risk level of ten, then, and we're doing estimation, our best case is going to take an hour.

Trevor Lohrbeer: Our worst case is going to take ten hours. Right. And so there is a risk level of 10. It's a 10 times spread in the amount of time. If I think it might take four hours, a risk level 10, it could take [00:50:00] from four to 40 hours. And so one way to kind of like just come up with your risk scores. Is to think what's my best case?

Trevor Lohrbeer: What's my worst case? And just divide my worst case over my best case and that gives me a kind of risk multiplier for my task Um, so I think that that can be a useful way to kind of think about risk The second thing that I add to that often is that people Will tend to be optimistic about their risk.

Trevor Lohrbeer: So having, um, categories of risk. So editing is for me is almost always a risk level of one because there's not much involved there. Whereas something like, um. Um, refactoring, rewriting, all that might be a level of two. Fairly easy, like, might take a little bit longer than I expect, but not dramatically longer.

Trevor Lohrbeer: Then we go into things like replicating, doing what [00:51:00] something, something someone else has already done, but I've never done it before. Well, maybe that's a risk level of three, you know, going up to things like designing whole systems. Maybe that's four. Invention might be five, might be ten for invention.

Trevor Lohrbeer: But, but like coming up with like, here's the different categories of tests I do, and assigning a default risk level to that. That you just automatically assume it's going to be that risk level and then you can tweak it from there But that way you're not going off and going like oh, yeah This is easily going to be like two When it's going to be designing a whole system, you're like no that really should be four or five, you know, um, and then Accepting that.

Trevor Lohrbeer: Yeah, if it means it's a a risk level of of five And I said it was you know, i'm estimating it four hours. I need to accept that it might be 20 hours You To do that test. So how do I factor that into my time blocking? And what that means [00:52:00] is I need to have a lot more less granularity in my time blocking.

Trevor Lohrbeer: So my time blocking is more I'm setting it creating a time block, which is advanced Project X forward, right? It's not down to the level of, well, it's, it's not a buffer. What it is, is just saying like, I'm going to allocate one, a lot more time blocks, but I'm not going to be as specific about what I'm doing in those time blocks, because I'm not going to say like, okay, I'm not going to say, okay, task one is time block is here's a two hour time block.

Trevor Lohrbeer: And then tomorrow I'm going to do test two in this two hour time block. And the next day test three and two hour time block. I'm just going to say, I'm working on project a. And I've got my list of tasks, and I'm working down that task list, recognizing that task 1 might actually take up all the time blocks in that week.

Francis Wade: Right. So this is, this is a three insight, three in one insight. [00:53:00] Who deserves three medals at this point. And I, and I, and I, you know, I'm looking at the clock and I'm thinking, Oh, so Trevor, Trevor, Trevor drops an insight bomb right at the end. We're crossing the finish line. Like, because we didn't have it all, sure we need it.

Francis Wade: So the first one I got was that the spread of your task, the distribution essentially, the, the earliest and latest times that we often assume that there's no spread, that if we think it's going to take an hour, it's going to take an hour. Really what we're seeing is going to take anywhere between a half an hour and five hours.

Francis Wade: And we don't account for that in our time blocking. We behave as if, you know, we've gotten the estimate is accurate with respect to the spread. And then we're over optimistic. So we think, oh yeah, I know when the median time is probably more like three or the average time is more like three and we don't account for that either.

Francis Wade: Um, and then the third thing you said is that, [00:54:00] The way we define the activity is we over define it, essentially, and say that we're going to do this part in this time block when the truth is, we don't know that we're going to do that part in that time block. We know that we're going to work on that project in that time block.

Francis Wade: If we become too specific, we run the risk of then. Not accomplishing what we said we were going to and that causes a problem Did I catch the did I catch that those three accurately?

Trevor Lohrbeer: Yeah, I think that's like I would make the one clarification there that um You can go more granular in your time blocking for lower risk tasks

Francis Wade: So like, you know, I I might

Trevor Lohrbeer: be able to say like hey I'm gonna like outline three articles in this hour time block because I know how long it takes me to outline an article Because i've done that like You know, a hundred times so I can go very granular.

Trevor Lohrbeer: Whereas once you go into tackling more, um, higher risk tasks, you have to go reduce your granularity. You have to look at and say, I'm not going to schedule this time block for a [00:55:00] specific task. I'm going to schedule it for a specific project and be at that higher level. And then the tasks I'm just going to try to make as much progress as I can in that time block.

Trevor Lohrbeer: You know, and you can do, you can do things like time block blueprints and things like that to kind of like push some of that a little bit forward, but the, the goal, the, the, the key there is that you're not going to be able to say, Hey, I'm going to be able to get task one done in a fixed amount of time because you don't know, because it has a high level of risk, a high level of time variability and how long it needs to get done.

Francis Wade: Right. So, so someone like an Evelyn then it's, it's, Evelyn is all of us at some point is that once you get past a certain level of. Simple time blocking, let's call it that, and you move into more complicated time blocking. The way you think about your schedule needs to change. The way you think about your task allocation to your canon needs to shift.

Francis Wade: The way you think about your estimations needs to alter. [00:56:00] And you need to have a, uh, uh, uh, a filter between simple tasks and more complex tasks. And you treat the two very differently. And you, even the sequencing of them actually no has an impact. If you're working on an overall complex project, it's not as simple as they're not all equal anymore.

Francis Wade: Um, Some have a greater impact than others and the actual sequencing now matters. Whereas, you know, when you're just cranking widgets, the sequence is like, I'll do whatever I feel comfortable doing in the

moment. So, but this is like, It's like another episode, um, Trevor, this is like a whole nother conversation for us to have.

Francis Wade: So you've teed us up for a continuation of this one, I think. Tell me if I'm wrong, but I don't see, I don't see us closing this out in the next two minutes and then saying, okay, we've, we've solved world hunger and we've solved the problem of, of hidden and complex tasks. What do you think?

Trevor Lohrbeer: Yeah, no, I [00:57:00] think we, we've got a whole nother episode in there.

Trevor Lohrbeer: So, um, just to summarize in there, I would say that if you're doing simple tasks, Managing your time works. If you're doing complex tasks, focus a little bit more on managing your risk and then managing your time from a, um, when you're using that, that time to move the progress forward, but not trying to do time estimation, um, if you've got high risk projects.

Francis Wade: I tell you, I, I want to talk to you about these things some more right now, but trouble, how can people hear more of your wisdom and your insights? You're such a, you're a polyglot of no mean order. Tell me if I'm wrong, but that's who I experience you. You're like brilliant in these different fields and how can they hear more of your wisdom?

Trevor Lohrbeer: Yeah, so I'm, uh, I blog regularly at dayoptimizer.com, which is also the digital day planner app that I run. I also [00:58:00] am running the podcast AI meets productivity. So if they go to [AI meets productivity.podbean.com](http://AI.meetsproductivity.podbean.com), or if the domain finally gets live, [AI meets productivity.com](http://AI.meetsproductivity.com). Um, you can also find that on any podcast player, um, it's in the Apple podcast store.

Trevor Lohrbeer: And, um, yeah, and then I occasionally blog at fastfedora.com, but that's been a little bit wild since then.

Francis Wade: Well, again, always a pleasure to have you on, um, love having you in this format because I know there's something that's going to happen and that's unprecedented and I just know I've never heard before and I'm pretty sure our listeners have, I just, we also hear these things in podcasts, these kinds of discussions or even in, in books for that matter.

Francis Wade: It's just. We just haven't gotten there when it comes to time blocking and task management in terms of the kind of the level of, say,

analysis, diagnosis that we need in order to be effective as we go forward.
[00:59:00] So Trevor, thanks so much for coming on the show. And folks, keep listening. There's more coming. I want to tell you about an episode that we have coming up.

Francis Wade: So keep listening.