

**Ep #35: Two Heads Are Better:
How Coworking Sessions Can Improve Law Firm Efficiency**



Full Episode Transcript

With Your Host

John E. Grant

[The Agile Attorney](#) with John E. Grant

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

In today's episode, I'm going to return to a concept that I first mentioned back in episode 15 when I talked about managing handoffs in your legal practice. And it's a technique that's common in the software world called paired programming. This technique is really consistent with one of the core tenets of agile, which is, we place a high value on individuals and interactions and that in-person real time communication is the best way to do that.

I'm going to describe at a high level how paired programming works in software teams. And then I'll go into some detail around how you can embrace the concept of paired programming or creating any kind of co-working or co-development opportunity to create better outcomes in your legal practice. Ready to become a more agile attorney? Let's go.

Welcome to *The Agile Attorney* podcast powered by Agile Attorney Consulting. I'm John Grant and I've spent the last decade helping lawyers and legal teams harness the tools of modern entrepreneurship to build practices that are profitable, scalable, and sustainable for themselves and their communities. Each episode I offer principles, practices, and other ideas to help legal professionals of all kinds be more agile in your legal practice.

Hey there, welcome back to the podcast. So, the past few weeks I have spent a good chunk of time talking about some of the counterintuitive elements of searching for efficiency. And if you remember, last week I talked specifically about how sometimes finding little efficiencies in your practice can actually work against some of your longer term goals or the overall flow through your practice.

And back in episode 33, I talked about how specifically looking for efficiencies in your client interactions can actually cause some downstream problems. This week I'm going to stick with this theme of efficiency. And I'm

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

actually going to introduce a technique that probably to a lot of people, is going to intuitively feel inefficient. And I'm going to make a case for why I think that it actually is something you can affirmatively do in a few different parts of your law practice. That even though at first it may feel like this is going to be less efficient or I'm spending too much time on this.

I'm pretty sure that if you stick to this practice, that you're going to actually find that a lot of your processes and interactions become more efficient in the long term. And the name of the process or the name of the technique is taken from the technology world and it doesn't quite translate on a one-to-one basis to legal, but the name of it is paired programming. And let me just describe a little bit what it means in software development, in the software world.

In order to do that, you have to understand just a little bit about the software development process. So, in any sort of new piece of code or new software that a team is developing, there's usually a design or a concept phase. And then there's going to be a specific phase where a computer programmer or software engineer is going to write the code for that program. And then after that, that code is going to get sent to a quality assurance phase.

And usually there's a quality assurance person or team or something that is going to actually run that code through a series of checks and make sure that it's actually working as designed. And also, that it's not breaking other parts of the tool or the software at large to make sure that it integrates well. Lots of different things and QA testing in software is a whole really mature practice. And there's some things to learn from that quality assurance process, I think in legal, that I will save for a future episode.

But the way that paired programming works is a little bit, again counterintuitive, which is, instead of handing the brief, the design, the particular need around a piece of code to a single developer. Where that

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

developer will sort of do their thing and write the code and then pass it off downstream to the next phase of the process. The software company or the team will actually have two developers working on that particular piece of code at the same time.

And it usually involves both of them sitting literally at the same desk or standing at the same desk or however it happens to be, with just one keyboard, one monitor, but they're both looking at the same thing. And obviously there's virtual ways to do this as well through screen sharing etc. But the idea is that instead of one person doing their thing and then passing it on to the next person. That it's going to be two people, although only one of them has control of the keyboard at any given moment.

And so, the general idea is the person who's writing the code is going to write the code, and obviously the other person can see what's being written but they should also be having a conversation during that process. So, the person with the keyboard is maybe talking out loud a little bit about I'm going to call this function or I'm going to use this array because I think that it's going to do X. And the other person is going to either sort of affirm that or maybe question that or maybe say, "Well, have you thought about this other function that we might call?"

And they're going to have those little micro conversations. And in the course of doing that, they're both going to be engaging with the problem in sort of this deeper way. And it's going to help them run through different possible solutions to that problem in a much more robust way. It becomes a generative conversation.

Another way is that obviously if someone is just going down a rabbit hole and they're not really talking out loud, then the other person, the observer, should be empowered to ask questions, "Hey, why did you decide to do it that way? Have you thought about this other thing?" And obviously in a world that doesn't have good psychological safety or camaraderie or

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

whatever, that stuff can come off as passive aggressive. So, there's some cultural things that you have to make sure are in place in order for this to be successful.

But when it is successful, you can kind of see how this is a two heads are better than one. Now, the pushback on that is, why would I pay two developers to be working on one thing at the same time? Why wouldn't I want to have each of them solving individual problems? Wouldn't it be more efficient to have each of them working on their own thing? And it comes back to how do you define efficiency?

If you're thinking of efficiency in terms of, how do I get some solution out the door to the most number of problems as quick as possible, then sure, having those two developers work on two different things might make sense. But when the things that you're doing have to actually work, have to be good at solving the client problems, and then also have to pass these QA tests, then you wind up actually getting a better solution more quickly because fewer things get rejected in the quality assurance, fewer bugs actually make it out into the wild.

And then also, obviously, well, hopefully at least, the software itself is more likely to solve the particular problems of the user that this piece of code is designed to solve. And to hearken back to one of the terms I used last week. If you're looking only in the local microcosm of your process, that is code development, then yes, paired programming is less efficient at delivering quantities of code.

But if we're looking at a system level in terms of delivering valuable solutions to our customers, then we actually get to a good and valuable solution more efficiently by allocating our resources in this way, even though it seems maybe inefficient in the microcosm. And there's some different manifestations of this concept of paired programming. And the one

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

I just described kind of assumes that both software developers have relatively equal skill sets, equal experience, etc.

But of course, there's lots of situations where they won't have equal skill sets. And so, you might pair a senior developer with a junior developer. And regardless of who's at the keyboard, that junior developer is going to be learning from the senior developer along the way. So maybe at first the junior developer is the person that is the observer. The senior developer's at the keyboard. But again, if the senior developer is talking out loud then maybe the junior developer is absorbing some of the information that they're saying.

Hopefully, again that junior developer has opportunities to ask questions. And then it might be true that the junior developer is familiar with different functions, different methods, different techniques. Maybe they got trained a different way, whatever it happens to be, that even though they're the junior developer, they might have some insight that the senior developer could find useful. So, it still is going to be a two way conversation. It just is going to be a little bit different dynamic as they go.

And then you can also imagine that those positions switch, that the junior developer is the person at the keyboard and the senior developer is the observer and it's the same sort of thing. The junior developer is talking out loud. The senior developer is reflecting on that, asking questions. But both of those situations are helping the junior developer get better. And they're also, again, helping ensure that the overall quality of the product remains high, that we are again, not going to have a bunch of rejections for bugs or for integration failures and that the end product is actually going to meet the needs of the customer.

And I think a lot of software companies that I'm aware of and even some that I've worked with from back in the day, find that paired programming is a really effective training method. But then again, even once the skill sets

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

sort of become more on par with each other, it's not like they use it only for training. It is a good technique for generating systemic efficiency and systemic quality even if we're not in that one-on-one training situation.

And there's a few reasons why that's true, one I hit on already, which is a lot of times, two heads are better than one. And even if that conversation and those maybe little micro disputes about what's the best way to do something, result in longer conversations, more time spent doing the code. It's okay if it does ultimately result in a better quality product. Something that is more likely to meet the need and less likely to have bugs in it.

The main reason though, why I think it's more effective is that a lot of those little disputes were probably going to happen at some point anyway. If the code failed QA and had to be sent back for fixes then there's going to be a dispute about whether it was the right way to do it anyway. And so, in having this be a conversation between two people and there's versions of paired programming where the quality assurance person is actually sitting in the writing section. So, there's different ways to sort of structure it.

But what we do is accelerate the feedback loop. And so instead of waiting for this piece of code to go to QA and then have some failure and have to come back to development to fix that failure. which means now the developer, maybe several weeks have passed and the developer has got to sort of reboot their brain around, what was I thinking when I called this function or when I wrote this particular piece of code? Which creates administrative overhead.

It's this non-valuable work that adds to the overall cost of developing the system but isn't really generating direct client value. It's sort of this rework ramp up whatever period. Related to that, it sort of reduces the opportunity for miscommunication, that game of telephone that happens where one person has done it one way and then they have to convey a message to

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

another person, who then has to convey a message to a third person or maybe it's a back and forth.

But whatever it happens to be, when those conversations are happening in real time, it's really easy for one of the people to say, "Hey, time out, you're using a word I don't understand or maybe I think we're thinking of this slightly differently. Can we make sure that we are on the same page before we go deeper into this discussion." And that sort of happening in real time helps make everything flow more smoothly. And again, get to that better outcome with one touch or one set of touches happening at the same time instead of having to do all of this back and forth.

And then the third way is that it just reduces the total number of handoffs between different players. And this is something that I talk a lot about in my presentations. I don't think I've talked about it much in the podcast yet, but handoffs are always, always, always a source of significant delay in any workflow that I've encountered. And that's because when you hand something off, it's rarely a one-to-one sort of that relay race where you pass the baton and the other person grabs it.

The thing that is more common is that you set down the baton and it goes into somebody else's queue, it goes into somebody else's inbox. And it takes a while for it to bubble up to the top of that person's time and attention and priority list. And so, every time a piece of work has to enter a queue, then you can be sure that there's going to be administrative delay and it's going to add to the overall cost. And therefore, sort of undermine the overall efficiency of the workflow that that piece of work is going through.

So how can we use that concept in a law practice? And there's a few ways. Number one is to do it really just like the software people are doing it. If you have to write a big complicated brief, if you have to dive in on a big complicated research project or whatever it happens to be, if you can do it

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

with another person in your firm, there's an opportunity to actually make the work product better.

And I'm not saying you can do this with everything, but I really do think that it can work in just that situation, where we're talking about peers working on something to get to a better result more quickly, a co-working session effectively. Of course, I think it can also work really well in the context of training. And I think that there are some misconceptions or again anti-patterns. I use that term a lot lately, in terms of how lawyers typically think about training their team members on how to do various things.

And what we want to believe is efficient is that we describe a process or a method or ideally even sort of have it written down, have a policy, a procedure, a manual on how to do a certain thing or set of things within your practice. And we train the new person on how we do that work. And then we expect them to go off on their own and be able to do that.

And I think another way of looking at it through that lens of paired programming is let's do it together the first time. And again, this isn't new to me. There's various different ways of expressing a you watch me and then I watch you. And then once I'm convinced that you're able to do this in a high quality and consistent way, then I'm going to let you do this on your own. And that can be a really effective way to train people up. It actually, I think again, is a more effective way.

And it certainly is going to be, I think, less frustrating and less time consuming over the long term than having that new person not quite understand. Or maybe your instructions weren't as clear as you thought they were and whatever the reason, they get it slightly wrong or maybe more than slightly wrong.

And therefore, you now, in your role as quality assurance or supervisor, have to engage either in all the work of sort of redlining and trying to show them where they missed the mark and then push it back to them, which is

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

creating these handoffs. So that's a lot of delay. That's a lot of time. Or you come to that situation where you're like, "Oh my gosh, I'm just going to do this. I don't have time to hand it back to them. I'll have to figure out how to train them on this. But I've got to get this out the door so I'm just going to fix it and send it down the way."

Which doesn't really create a very good feedback loop for your new person and obviously can create a lot of frustration for you. There is sort of a half step that you can do that I just want to mention briefly. We actually talked about it, Alex Devendra and I talked about it back in episode 12. And then it came up again with Justie Nicol in episode 30, which is using Loom.

When you're reviewing somebody's work, if you can't do it sort of with them in that paired programming sort of way, you can still create some of that experience by turning on your screen recorder and your microphone. And then as you're reviewing somebody's work, you're talking out loud about what it is that you're seeing or not seeing or need to see in terms of helping somebody understand that a little bit better. And I've got a few different clients that do that on occasion with their team members and the team members seem to really like it.

It is a version of personal feedback, but it doesn't have to happen in real time. It can be time shifted in a way that you can record it and they can consume it on a schedule that works for you.

Another way to adopt this notion of paired programming in law practice, and I actually talked about this back in episode 15. And I think I introduced the concept of paired programming back then too. But there's this meme that has been popular on LinkedIn and sort of other corporate culture, I guess, I don't know, parts of the internet. Where basically people say, "This meeting could have been an email." And it's counter to the idea that meetings can often waste a lot of time and why not just summarize it in an email? I often think the opposite is true.

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

I think that you can spend a lot of time trying to offer feedback or insight or things via an email. When really you should just pick up the phone and call somebody or call a meeting and sit down face-to-face with them, which is the best possible situation, and actually talk it through. Because you're never communicating quite as well in email as you think you are or at least as you intend to be.

And again, I think this is really important and impactful when you're working with members of your team in terms of doing that quality assurance or giving feedback or again, even drafting from the get go to be sort of in that pure paired programming situation.

Another place, and this gets back to some of the things I talked about a couple of episodes ago around client engagement is, I think it can be a really effective tool in terms of working with clients to get their homework information back from them. And there's a technique that I suggest to a lot of my clients when we're talking about client homework, which is, yes, go ahead and assign the homework. Hopefully when you do assign that homework, you're doing it in a real time conversation and not just an email.

We want to follow up with email so that we've got the record and the documentation and the client's got something to jog their memory. But I usually think that it's better to sort of, at least assign that homework. Again, whether it's a request for information or for documents, or some instructions about how to review whatever document it is that you're preparing for the client, that they need to review. We want to initially start that in a one-to-one conversation.

Have the email, give them a deadline, which is a whole different topic that I think I've hit on a little bit before. But then my favorite way to reinforce that deadline is to schedule a follow-up meeting. And if you've given someone a five day deadline or a one week deadline. I want to schedule a meeting that corresponds with that deadline. And it may just be a quick check-in, a

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

phone meeting, whatever. And I will often suggest that you tell the client a few things.

Number one, you don't get to cancel this meeting unless you have completely finished whatever assignment I've just given to you. So, if the one week passes and you've delivered all of the documents or given me all the information I need and you don't have any questions and I don't have any questions, then we're going to cancel that meeting. If you've given me the documents but I have questions or you maybe didn't give me everything or you have questions, whatever it happens to be, we're going to hold that meeting.

And we're going to close the feedback loop right then and there to make sure that you know what I'm asking for. I know what your issues are. I understand if maybe I didn't explain the assignment in a way that resonated with you or that felt actionable to you. Whatever it happens to be, we're going to have that meeting either way.

And it may even be that we're going to have that meeting and instead of you asking the questionnaire or answering the questionnaire questions that I sent you via email. I'm going to ask you the question and I'm going to type while you talk or I'm going to record it or I'm going to turn on my voice to text transcriber anymore and we're going to capture your answers that way. What I like about that is it makes sure that things sort of stay on the front burner. That not too much time passes before you're getting the information you need so that you can push the matter forward.

And it winds up being a really effective technique. Even if the client does none of their homework at all, then it still allows you to engage with them. And if they show up completely unprepared for that meeting and you say, "Well, when can you get this to me?" And they're like, "I don't know." Well, then that gives you information you need about whether you want to continue that engagement.

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

Because you don't want to be in a position where you're spending your or your team members' cycles having to basically be a nag to chase down, to bird dog all of this stuff. You want to work with clients that are engaged and motivated to help themselves.

So hopefully I've given you a few ideas, maybe a little inspiration to think about ways that are going to help you in your law practice, get to a higher quality result with less administrative overhead, less handoffs, less time in queue, less miscommunication. And I think really pushing towards more of these paired programming or working session type situations, where we're engaging in real time conversation, even though it feels a little bit inefficient.

And even though, maybe there's some ways like I mentioned with Loom to do the time shifting. And email of course is the ultimate time shifter where I can write an email whenever I have time and then you can answer an email whenever you have time. But we all know that that doesn't work especially well. And so, we want to try to compress these timeframes, improve and accelerate the feedback loops. So that we're getting to that higher quality outcome, that higher quality work product in a faster and ultimately more efficient way.

Thanks for listening to *The Agile Attorney* podcast. I'm your host, John Grant. If you found today's episode interesting or useful, please share it with someone who you think would benefit from a more agile approach to their legal practice. If you have any questions, feedback or maybe a topic you'd like to hear me cover, you can reach me at john.grant@agileattorney.com.

To help other attorneys and legal professionals discover this podcast, it helps a lot if you could rate or review me on Apple Podcasts or Spotify. And of course, be sure to subscribe in your favorite podcast app. This podcast gets production support from the fantastic team at Digital Freedom

Ep #35: Two Heads Are Better: How Coworking Sessions Can Improve Law Firm Efficiency

Productions and our theme song is the instrumental version of Hello by Lunareh. That's it for today's episode. Thank you for listening and see you next time.