

September 20, 2009

PREOCCUPATIONS | JIM REMSIK

For Writing Software, a Buddy System

By PATRICIA R. OLSEN

I'M a programmer at Hashrocket, a Web development firm in Florida. Our style of working is called pair programming, which has been popular for years in some software design companies. Two of us sit side by side at a computer workstation to develop a program that is the backbone of an interactive Web site.

One person does the actual writing, or coding, and the other person checks it, corrects it and offers suggestions as it's being written. Programmers, or software developers, refer to these roles as driver and navigator.

It might sound as if the person writing the programming code would find it distracting to work this way, but it's not. It's a collaborative effort, and that's the beauty of it. Proponents believe it saves a company time and money. Bugs can be found more quickly, and the code is written more efficiently when two people create it simultaneously. In this case, two heads are definitely better than one.

Consider the game "Where's Waldo," in which a cartoon character is hidden in an intricate design. Most people can eventually find Waldo after poring over the drawing. Similarly, when programmers check code for errors, it takes time to examine the logic and find mistakes.

Now imagine if someone sat next to the artist from the very beginning. Obviously, the onlooker should be able to find Waldo more easily. The character would stand out. In the same way, one programmer looking for errors in code as another writes it can follow the logic in real time. Ideally, the navigator immediately catches anything that is incorrect. My colleagues agree with this analogy.

Part of our job is to attend design meetings with new clients so we understand what they want, but when we're developing software programs we work in pairs 100 percent of the time. Teams are assigned at our daily 9 a.m. meeting.

We try to work together at least one full day and we often spend several days together. We switch roles, too, and we constantly change partners, which is called promiscuous pairing. People have different talents, and this way the expertise is spread around.

To me, pair programming is the only way to work. Writing code is not only scientific, it's also a creative process. I get writer's block sometimes. To be able to collaborate with someone is great.

When senior and junior developers are paired, junior programmers might feel intimidated. If this happens, the junior programmer might be asked to start as the driver, which may encourage the senior person to become a better teacher. It's also a way to bring junior programmers up to speed quickly, because they benefit from the more senior employee's knowledge.

When programmers interview for a job here, we ask that they spend a week with us, and we pair them with as many people as possible. Besides checking their skill level, we want to see if we could all work side by side with them for weeks at a time and if they're a fit with the company. It gives them a chance to check us out, too.

Our programmers exhibit idiosyncrasies that might be found in any workplace, such as one person leaving a work area messy, and the germaphobe who disinfects everything. On certain days, people may not be the best partners, like the day I was preoccupied because my dog was at the vet. If that occurs, the other person can usually snap their partner back to center line.

People working so close have occasional personal conflicts, too. Hashrocket provides our workspace, office furniture, mouse pads and anything else we need, but we use our own laptops.

Once two of our programmers had a falling out over a keyboard function. The navigator wanted to remap the caps lock key as a control key for when they switched roles. But it was the driver's laptop, and he didn't like that idea. When the driver went to the restroom, the navigator changed the key anyway, and the two had a disagreement. They came to terms when the navigator agreed that the person who owns the laptop gets to decide the computer key function.

The ability to pair program all the time was a selling point for my taking this job. Previously, I worked for an I.T. group at a government agency in another state. I tried to get buy-in for pair programming there, but we were allowed to do it only occasionally because of the perceived drawbacks. State governments have tight budgets, so it was a hard sell. Some people can't believe it saves money for two people to do a job one person can do.

THIS style of programming all day every day is exhausting, but it makes me more disciplined. It's easy for a programmer working alone to give in to distractions — to check e-mail, [Twitter](#) or [Facebook](#), for example — when confronted with a problem. We're on social media a lot anyway, because we need to stay current, but many people take five-minute breaks every half hour, and that's the recommended time for accessing Twitter or the Internet.

That schedule of working 25 minutes and then taking a 5-minute break is a time management technique called Pomodoro, which means "tomato" in Italian. Every once in a while a programmer won't stop after 25 minutes and we'll hear someone say, "Respect the tomato."

As told to Patricia R. Olsen.

Copyright 2009 The New York Times Company

[Privacy Policy](#) | [Terms of Service](#) | [Search](#) | [Corrections](#) | [RSS](#) | [First Look](#) | [Help](#) | [Contact Us](#) | [Work for Us](#) | [Site Map](#)