



Robert C. Martin Series

The Software Craftsman

Professionalism, Pragmatism, Pride

Foreword by Robert C. Martin

Sandro Mancuso

The Software Craftsman

probably not as good as we think we are. A good developer is a developer who can write code that any other developer can understand. When our pairs don't agree or don't understand what we are doing, we should take this as an opportunity to have a good discussion. Use it to learn something new and open your mind to different approaches. If someone is questioning what we've just done, maybe it is because it's not good enough and there is a better way of doing it. We should take opportunities like that to share what we know, making everyone around us better. When teaching, we are forced to structure our thoughts, making us really understand our quite often half-baked ideas so we can make someone else understand them.

Pairing with someone from our team or a friend is great, but pairing with someone that we barely know can be even better. Usually team members and friends, after some time working and pairing together, develop a common understanding and style of coding. When pairing with people we have never paired with before, we end up potentially exposing ourselves to very different ways to solve and think about problems. The best way to find different pairing partners is attending meetings organized by our local user groups or technical communities. There are also an increasing number of developers willing to set up remote pair-programming sessions in whatever we want to work with. There are plenty of tools out there that can make a remote pairing session very smooth.

We need to keep our minds open to new ideas when pairing. Sometimes we learn, sometimes we teach, and sometimes we do both.

SOCIALIZE

Not only individuals and interactions, but also a community of professionals.

The idea that software developers are introverted nerds is totally outdated. Finding other developers whom we can bounce ideas off of, pair-program, and network is almost essential for a successful career. A great and easy way to do that is to join your local user groups and technical communities and to participate in their events. They normally promote free talks, coding activities,

and social events. Another great aspect of being part of a community is the feeling that we are not alone. User groups and technical communities tend to be extremely open and welcoming. We find developers from many different backgrounds, working for completely different industries, with different technologies, languages, and processes. Some developers are more experienced than others but there is one thing they all share: passion. The great thing about passionate developers is that they are constantly learning and are very happy to share what they know.

Being a member of a local user group or technical community is a fantastic way to learn and share ideas.

DELIBERATE DISCOVERY

I'm the smartest man in Athens because I know that
I know nothing.
—Socrates

The biggest mistake that software professionals can make is not accepting that they don't know what they don't know. Not knowing what we don't know is also called *second-level ignorance*. Accepting that we have a lot to learn is a sign of maturity and one of the first steps toward mastery.

The vast majority of us have a natural tendency to be positive and optimistic. An example of that is how bad we are at estimating tasks. Once tasks are completed, if we compare the amount of time they took to our original estimations, we will see that the majority of them took longer than we expected. We need to accept that there is a massive chance that things will not go according to plan, which means there will be unforeseen and unpredictable problems. Unfortunately, we have absolutely no idea when, where, or how. The consequence of us ignoring this fact is that we will be caught by surprise and will not be able to handle the problems as well as we could if we knew about them upfront.

There is not a magical way to completely solve this problem but we can try to minimize it. One way of doing this is to constantly expose ourselves to situations

where we can learn something new about the context we are in. This is very important mainly in the early days of a project or before building a major set of new features—when we are most ignorant about what we need to do. Spending time trying to identify and minimize our ignorance across all the axes we can think of can be time extremely well spent.

Ignorance is a constant. Imagine we could start our latest project from scratch again. Same people, same requirements, same organizational constraints, same everything, but the difference this time is that we would start with all the knowledge we already have. How long do you think it would take? Now, stop and think about it. When asking this question, usually the answers average between one half and one quarter of the original time and that's where my own answer would be as well. Ignorance is the single greatest impediment to throughput, meaning that if we reduce the level of our ignorance as fast as we can, we can be far more productive and efficient.

We should always try to create opportunities where we can learn something we don't know. "But if I don't know what I don't know, how can I create opportunities to learn that?" you may ask. Speak to random colleagues and ask them how they keep up with the progress in our industry. Go to technical community and user group events. Show your code to other people. Ask for help even when you think you don't need it. Try to figure out which aspects of your current project you and your team have not explored yet, then start discussions about it or even write a proof of concept. Aiming to remove the ignorance constraint should always be your priority in order to deliver projects more efficiently and grow as professionals.

WORK-LIFE BALANCE

So far I have been saying how important it is to look after our careers and that we should dedicate loads of time outside working hours to practice and learn. However, we all have family, friends, and other interests in life. I have a wife and two kids, whom I love to bits and want to spend a lot of quality time with. It is never easy to balance work and personal life because we spend more than 50 percent of our waking hours at work, including all the time commuting to and from it.

The most common thing I hear every time I talk about investing time in our own careers outside working hours is, “I don’t have time.” And if you say or agree with that, you are probably right. That’s what you decided to believe and that, in turn, became your reality. But the truth is, we all have time. We are just not very good at optimizing it. Maybe we prefer to spend our time with something else that may or may not be as important as our careers.

Stop reading now and think about what you did yesterday, from the time you woke up to when you went to bed. What did you do the day before yesterday? Seriously. Stop and really think about that. Now think about how much of it was waste? How much of it was productive? By productive we can mean many things, from learning something new to spending time with our loved ones. If you took some time to rest, this is also important, especially after a busy day or a busy week. Our bodies need to recharge and we should cater to that as well. Some people genuinely do not have a lot of time. I once met a German speaker at a conference who told me he had a wife and five kids. Talking about learning and practicing outside working hours, he told me that his only alternative was to be very smart in how he uses the very limited amount of time he has. Despite his tough situation, he managed to present at a conference.

CREATING TIME

Quite often, lack of time is used as an excuse for our own laziness. People are different. Some live in large cities, some in small villages, some have family, and some live on their own. Some people have hobbies. Some love going to the gym. Some are young and love going out with friends; others are older and prefer to stay at home. Some are morning people; some go to bed very late and hate to wake up early. By no means do the following tips apply to everyone; they are just a collection of things that we can do to create time and use it to invest in our careers.

I realized that I used to waste a lot of time in front of the TV, aimlessly browsing the Internet, checking all the uninteresting things my friends publish on social networks, playing computer games, or watching sports. I decided to cut down on the number of hours spent on these things. That doesn’t mean I don’t do them anymore. It just means that I do them in moderation. Although they are great ways to relax and switch off, we do not need to do them every night, all night.

Your local coffee shop is your friend. Find a coffee shop near your workplace with a good Internet connection. If there isn't one, find one along the way. Even if your company has a coffee area, I would avoid that since we can be tempted to do work during this time or may be interrupted by colleagues. Take one day a week and try to get to a coffee shop one or two hours before you start working. Use this time to write code, read a technical book, blogs, or whatever you think you need to do to learn and move your career forward.

Lunchtime is another great opportunity we have to practice and learn something new. Once or twice a week just grab a sandwich and your laptop, and go somewhere quiet. It is amazing what we can do in such a small period of time when we are focused.

Check if there is a user group or tech community in your city and join them. Usually user groups meet regularly; many operate on a monthly schedule. Make a commitment to go to the meetings at least once or twice a month. Meetings tend to last between one to three hours so I'm sure the vast majority of us can plan for that. The advantage of attending user group meetings is that we usually can learn a huge amount in a very short period of time, either from a presentation or from speaking or coding with other developers. There is a limit to what we can learn on our own. Although we all feel we can learn anything, sometimes we don't know where to start. It's also worth remembering that it can be a lot faster to learn something from someone with more experience than trying to learn on our own.

Go to bed 30 minutes earlier than normal and use this time to read a book, look at blogs, or watch technical screencasts before falling asleep. This is something that works really well for me. Every single night I try to read at least a few pages, regardless of the time I go to bed.

Buy yourself a Kindle, iPad, or another e-book reader, and carry it with you at all times. Use it every time you have some dead time, like commuting or waiting for your dentist, doctor, or hairdresser appointment.

I believe every developer has a good smartphone today but if you do not have one, buy one now. Use Twitter or any other information aggregator when you

have a break or dead time. This is a great way to quickly read something or keep up with the latest news and trends. Being able to go to a single place and get all the information you want is key when you want to optimize time.

FOCUS: THE POMODORO TECHNIQUE

In order to use our time outside working hours wisely, it is extremely important that we focus. A good technique is to decide beforehand what we want to do before we actually use this time slot. It is like deciding the agenda before scheduling a meeting. It does not need to be strict, but we need at least to have a good idea of what we want to achieve. Once this is done, we must ensure we can focus and get it done. One way we can achieve this is by using the Pomodoro technique. There are five basic steps to implement this technique:

1. Decide on the task to be done.
2. Set the Pomodoro (timer) to 25 minutes.
3. Work on the task until the timer rings.
4. Take a short break (normally 5 minutes).
5. Every four “Pomodoros,” take a longer break (15–30 minutes).

During a Pomodoro (25 minutes), we focus on the task and nothing else. The breaks between Pomodoros are for a quick rest, coffee, checking emails, making a phone call, checking your Twitter, or whatever you feel like. We should do whatever we can to finish the Pomodoro with no interruptions but in case it needs to be interrupted (there is an important call we need to take or we really need to speak to someone), then the Pomodoro must be terminated and not paused. A new one should be created when we are ready to work on the task again. There are many Pomodoro tools available out there. Some are very sophisticated where you can keep track of all the tasks you completed, interrupted Pomodoros, and many other statistics. I, personally, prefer the simple ones but feel free to use one that suits you better.

BALANCE

Whatever you do, a sustainable pace is key. Keeping a healthy work-life balance is tough but not impossible. If you are the type of person that says, “I don’t want to