11. Pressure

Imagine that you are having an out-of-body experience, observing yourself on an operating table while a surgeon performs open heart surgery on you. That surgeon is trying to save your life, but time is limited so he is operating under a deadline—a *literal* deadline.

How do you want that doctor to behave? Do you want him to appear calm and collected? Do you want him issuing clear and precise orders to his support staff? Do you want him following his training and adhering to his disciplines?

Or do you want him sweating and swearing? Do you want him slamming and throwing instruments? Do you want him blaming management for unrealistic expectations and continuously complaining about the time? Do you want him behaving like a professional, or like a typical developer?

The professional developer is calm and decisive under pressure. As the pressure grows he adheres to his training and disciplines, knowing that they are the best way to meet the deadlines and commitments that are pressing on him.

In 1988 I was working at Clear Communications. This was a start-up that never quite got started. We burned through our first round of financing and then had to go for a second, and then a third.

The initial product vision sounded good, but the product architecture could never seem to get grounded. At first the product was both software and hardware. Then it became software only. The software platform changed from PCs to Sparcstations. The customers changed from high end to low end. Eventually, even the original intent of the product drifted as the company tried to find something that would generate revenue. In the nearly four years I spent there, I don't think the company saw a penny of income.

Needless to say, we software developers were under significant pressure. There were quite a few very long nights, and even longer weekends spent in the office at the terminal. Functions were written in C that were *3,000 lines long*. There were arguments with shouting and name calling. There was intrigue and subterfuge. There were fists punched through walls, pens thrown angrily at whiteboards, caricatures of annoying colleagues embossed into walls with the tips of pencils, and there was a never ending supply of anger and stress.

Deadlines were driven by events. Features had to be made ready for trade shows or customer demos. Anything a customer asked for, regardless of how silly, we'd have ready for the next demo. Time was always too short. Work was always behind. Schedules were always overwhelming.

If you worked 80 hours in a week, you could be a hero. If you hacked some mess together for a customer demo, you could be a hero. If you did it enough, you could be promoted. If you didn't, you could be fired. It was a start-up—it was all about the "sweat equity." And in 1988, with nearly 20 years' experience under my belt, I bought into it.

I was the development manager telling the programmers who worked for me that they had to work more and faster. I was one of the 80-hour guys, writing 3,000-line C functions at 2 AM while my children slept at home without their father in the house. I was the one who threw the pens and shouted. I got people fired if they didn't shape up. It was awful. I was awful.

Then came the day when my wife forced me to take a good long look in the mirror. I didn't like what I saw. She told me I just wasn't very nice to be around. I had to agree. But I didn't like it, so I stormed out of the house in anger and started walking without a destination. I walked for thirty minutes or so, seething as I strode; and then it started to rain.

And something clicked inside my head. I started to laugh. I laughed at my folly. I laughed at my stress. I laughed at the man in the mirror, the poor schmuck who'd been making life miserable for himself and others in the name of—what?

Everything changed that day. I stopped the crazy hours. I stopped the high-stress lifestyle. I stopped throwing pens and writing 3,000-line C functions. I determined that I was going to enjoy my career by doing it well, not by doing it stupidly.

I left that job as professionally as I could, and I became a consultant. Since that day I've never called another person "boss."

Avoiding Pressure

The best way to stay calm under pressure is to avoid the situations that *cause* pressure. That avoidance may not eliminate the pressure completely, but it can go a long way towards minimizing and shortening the high-pressure periods.

Commitments

As we discovered in Chapter 10, it is important to avoid committing to deadlines that we aren't sure we can meet. The business will always want these commitments because they want to eliminate risk. What we must do is make sure that the risk is quantified and presented to the business so that they can manage it appropriately. Accepting unrealistic commitments thwarts this goal and does a disservice to both the business and to ourselves.

Sometimes commitments are made for us. Sometimes we find that our business has made promises to the customers without consulting us. When this happens we are honor bound to help the business find a way to meet those commitments. However, we are *not* honor bound to *accept* the commitments.

The difference is important. Professionals will always help the business find a way to achieve its goals. But professionals do not necessarily accept commitments made for them by the business. In the end, if we can find no way to meet the promises made by the business, then the people who made the promises must accept the responsibility.

This is easy to say. But when your business is failing, and your paycheck is delayed because of missed commitments, it's hard not to feel the pressure. But if you have behaved professionally, at least you can hold your head high as you hunt for a new job.

Staying Clean

The way to go fast, and to keep the deadlines at bay, is to stay clean. Professionals do not succumb to the temptation to create a mess in order to move quickly. Professionals realize that "quick and dirty" is an oxymoron. Dirty always means slow!

We can avoid pressure by keeping our systems, our code, and our design as clean as possible. This does not mean that we spend endless hours polishing code. It simply means that we don't tolerate messes. We know that messes will slow us down, causing us to miss dates and break commitments. So we do the best work we can and keep our output as clean as we can.

Crisis Discipline

You know what you believe by observing yourself in a crisis. If in a crisis you follow your disciplines, then you truly believe in those disciplines. On the other hand, if you change your behavior in a crisis, then you don't truly believe in your normal behavior.

If you follow the discipline of Test Driven Development in noncrisis times but abandon it during a crisis, then you don't really trust that TDD is helpful. If you keep your code clean during normal times but make messes in a crisis, then you don't really believe that messes slow you down. If you pair in a crisis but don't normally pair, then you believe pairing is more efficient than non-pairing.

Choose disciplines that you feel comfortable following in a crisis. *Then follow them all the time*. Following these disciplines is the best way to avoid getting into a crisis.

Don’t change your behavior when the crunch comes. If your disciplines are the best way to work, then they should be followed even in the depths of a crisis.

Handling Pressure

Forestalling, mitigating, and eliminating pressure is all well and good, but sometimes the pressure comes despite all your best intentions and preventions. Sometimes the project just takes longer than anyone thought it would. Sometimes the initial design is just wrong and must be reworked. Sometimes you lose a valued team member or customer. Sometimes you make a commitment that you just can’t keep. Then what?

Don’t Panic

Manage your stress. Sleepless nights won’t help you get done any faster. Sitting and fretting won’t help either. And the worst thing you could do is to rush! Resist that temptation at all costs. Rushing will only drive you deeper into the hole.

Instead, slow down. Think the problem through. Plot a course to the best possible outcome, and then drive towards that outcome at a reasonable and steady pace.

Communicate

Let your team and your superiors know that you are in trouble. Tell them your best plans for getting out of trouble. Ask them for their input and guidance. Avoid creating surprises. Nothing makes people more angry and less rational than surprises. Surprises multiply the pressure by ten.

Rely on Your Disciplines

When the going gets tough, *trust your disciplines*. The reason you *have* disciplines is to give you guidance through times of high pressure. These are the times to pay special attention to all your disciplines. These are *not* the times to question or abandon them.

Instead of looking around in a panic for something, anything, that will help you get done faster, become more deliberate and dedicated to following your chosen disciplines. If you follow TDD, then write even more tests than usual. If you are a merciless refactorer, then refactor even

more. If you keep your functions small, then keep them even smaller. The only way through the pressure cooker is to rely on what you already know works—your disciplines.

Get Help

Pair! When the heat is on, find an associate who is willing to pair program with you. You will get done faster, with fewer defects. Your pair partner will help you hold on to your disciplines and keep you from panicking. Your partner will spot things that you miss, will have helpful ideas, and will pick up the slack when you lose focus.

By the same token, when you see someone else who's under pressure, offer to pair with them. Help them out of the hole they are in.

Conclusion

The trick to handling pressure is to avoid it when you can, and weather it when you can't. You avoid it by managing commitments, following your disciplines, and keeping clean. You weather it by staying calm, communicating, following your disciplines, and getting help.