Our Best Hope

Forrest Shull

WHEN I THINK about practical techniques that help members of software development teams truly be “reflective practitioners,” my mind goes straight to project retrospectives. Our special issue wouldn’t be complete without discussing this approach, which lets teams learn from their own experiences, reflect on what has actually happened, and see how to do it better next time. To learn more, Linda, I got interested in retrospectives initially because of you and by participating in the various retrospectives that you facilitated for our annual editorial board meetings. But how did you get interested in retrospectives in the first place?

I was working at a medium-sized telecom company, and my job as part of the research organization was to help teams work with patterns. There was a very large project that kept growing and growing, with ever more money being invested. After two years, the project got cancelled and was an enormous and very expensive failure. My boss came to me and said, “I know that you’re working on patterns, and I know they have to do with architecture, design, and other technical things... but surely there are patterns for how you run projects, how teams do what they do. What we want to do is to make sure we get hold of some of those, so that we don’t see this kind of massive failure again. Can you find some?”

I had a good friend who was writing a book about retrospectives. That was Norm Kerth, who wrote the classic book on the subject, Project Retrospectives (Dorset House, 2001). I thought that if anybody knew how to make sense out of what projects were doing, and if there were patterns, it would be Norm. I contacted him, and he said I should do a retrospective. I didn’t know what that was or how to do it, so he did a great job of holding my hand and guiding me through the retrospective wilderness and giving me some ideas for exercises. In going through that initial retrospective, I learned an enormous amount. It was very informative and helped me in the way I see teams, patterns, and retrospectives, and what people can learn from their experiences.

It helped in the way I see teams, patterns, and retrospectives, and what people can learn from their experiences.

I spoke with Linda Rising, an independent consultant who is internationally known for her work on agile practices, affecting change in organizations, and retrospectives.

Of course, our discussion resulted in more of Linda’s great stories than I could include here, so if you’d like to hear the interview in its entirety, please visit www.computer.org/software-multimedia.

IEEE Software Mission Statement

To be the best source of reliable, useful, peer-reviewed information for leading software practitioners—the developers and managers who want to keep up with rapid technology change.
ity, the last check on the list of things the projects had to do. Nobody on a succeeding project ever went back, looked at that list, and said, “Now what can we on our project learn from that other one?”

Since you bring up postmortems, I think, at heart, both postmortems and retrospectives have very similar goals. Is there an easy way to characterize the main difference between the two?

Absolutely. Postmortem is Latin for “after death,” and it comes from the medical domain—the idea of doing an autopsy to investigate what caused someone to die. Of course, we’d like to learn from it if we can, but there isn’t anything you can possibly learn from a postmortem or an autopsy that can help the dead person. That life is done; it’s over. The idea was transferred to projects: the project is dead, the project is over, so there isn’t anything we can learn from the experience that can help that project.

In a retrospective, the idea is more like a check-up, if we want to stick with the medical analogy. You’re going to sit down with your physician and say, “You know, I’ve been having some chest pains.” The doctor will come up with some ways to improve your life. It’s about learning. It’s about stopping and thinking about what you’re doing. That idea transfers well to a project: let’s take a moment to stop and think about what we’re doing, with the idea that maybe we need some more vitamins, maybe we need some exercise, maybe we need to learn about how to tweak things while we still can.

In your experience of working with so many teams, how receptive are they to this idea? In my experience, any time you’re asking people to take time away from “productive work” to be reflective, it’s a hard sell. People tend to be under such pressure to get things done.

I like to use a sports analogy. People can appreciate the difference in performance in a sports team after having a huddle in the middle of a game, where the coach might come out and say, “Here’s how we’re doing, and I think we need to adjust our game plan. We were trying to use a particular set of plays and that’s not working because the other team is doing things that are different from what we anticipated. We have to do some replanning on the fly.” They don’t have time for an enormous meeting, where they lay out a whole complicated game plan, but they do have time for tweaks in the way things are going that are based on what they’ve seen so far.

Teams buy into that, as long as it doesn’t take a long time and as long as something real happens. It can’t just be a meeting; it can’t be, “Let’s sit and discuss interminably.” Any adjustment has to be proactive, specific, and a targeted action. But not too big! Tiny things are within our locus of control.

In your role as a facilitator and educator with teams applying retrospectives, can you get a sense of when teams are really getting it? That is, when the retrospective is a real learning opportunity versus going through the motions?

Yes! In the teams I work with, which are now mostly some flavor of agile, the idea is that you don’t come into a room for a meeting where you’re going to sit and brainstorm. We’re moving more toward continuous retrospectives. Everything is continuous now; continuous...
delivery, for instance. So retrospectives need to match that. The idea is not that every four to six weeks, we sit down for an hour or so and come up with some action plans. No! We have it going on in the backs of our minds all the time. We do that with an ongoing timeline, which is basically a big board that’s divided up into chunks of time. As events happen, people post cards on the timeline, reflecting the event and how they thought about it. People see that, all the time. They go by and read it, and they see when new cards are posted.

This does a couple of things. It gets the idea, the event, and the feeling out into the open, and other people notice it, and can share it and talk about it. It also starts everyone thinking about what can be done differently to address the problem that this event calls to our attention.

One of the things we know is that the human brain has levels of processing, and we can tuck things away for a later meeting when we’re going to talk about it, but in the meantime, we can just let it grind away subconsciously. When the meeting actually happens, people have already been cranking away and thinking about ideas, maybe for the entire iteration. When they come in to the retrospective, the meeting start is like, “Hey, how about this? I’ve been thinking about that problem we had with the customer, where they said they wanted the red box, but now they changed their minds and it’s going to be a green box. You know, if we had done...” Or they’ll have an idea. The excitement in a meeting like that is palpable. These ideas aren’t hard-and-fast things that will definitely work. Each one is an experiment. We’ll try it for the next iteration, and we’ll see how it works. And we’re not going to do too many experiments, just one or two.

For me, a retrospective is a time when we look back over the last iteration, at the experiments we tried, and we say what worked really well, what we should try differently next time, and what we learned from these experiments. We incorporate that into the planning for the next couple of experiments that we’re going to try in the next iteration. It’s an ongoing process. It’s not necessarily about identifying all the ways that we can improve. Of involvement over to people, they get excited, and they think, “Well, maybe I can do something about this.” And you can pick that up. You can definitely feel that.

It’s different from an ordinary retrospective, where you’ve got a lot of people coming in, and a lot of people are late, and you say, “We’re going to start brainstorming and think about what worked well and what we should be doing differently...” But it just doesn’t have any juice, it doesn’t have any energy, and people are bored and think, “Nothing is going to happen. Who cares? We’re just wasting time here. Nobody listens to us anyway.”

Do you have any recommendations for what to do if you’re a facilitator, and you’re in one of those situations where there isn’t a lot of hope or energy? Do you have ways to kick things out of that mode and into a more positive direction?

There always has to be a first meeting. In the first meeting, we have to realize that as bad as things are, there’s always something, some small thing that we can start with. But really, there are no small things. Everything can have a wide impact. You don’t know, in the beginning, whether your small idea is going to have enormous impact or whether it’s going to lead to another small thing, and then another small thing, and that will have enormous wide impact. You don’t know, so you have to keep coming up with little things. It doesn’t have to be ginormous, just some little tweak, and that will get the ball rolling.

Of course, it shouldn’t be just about improving this team. You should periodically get together with everybody else in the organization and share what you’re tried. What
you find is there’s more commonality across projects than you realize.
If you do get in the business of trying to share what you’ve learned with
other teams, they’re really interested in that. They’re really interested to
hear what your experiments were, because you’re talking about little things that they could possibly do.

When the same people (with the same biases) doing the retrospective
are the ones doing the work, how do you get people to adopt a new
perspective and see things from the outside?

Well, if it’s possible, include a customer or representative from whichever organization is certifying your work in the retrospective. You could ask them, “What are some tweaks we should try for improvement?” In fact, maybe you could identify ways to work together.

I have a story that I tell about a customer; this was that same tele-
com company I mentioned earlier. What actually started us doing agile was that the telecom industry was changing underneath us. The customer came to us and said, “We don’t know what we’re going to need, but we know we’re going to need it by June.” Our team said we could do that within four weeks, and then we would get together again and see what else we knew, because in four weeks everybody will have learned something about the marketplace, about how this system is going to work, about our competitors and what they’re doing. We knew we would know a little more, and then we could meet and decide what would be the next baby step. When we got to the end (and we did deliver on time, by the way) the retrospective included comments from the customer, saying, “In the beginning, we didn’t know anything, but we grew this together.”

Are there general antipatterns that people tend to fall into that make the process less efficient than it could be?

One of the problems is that we’re stuck with our brains, and there’s nothing we can do about that. Human brains have an enormous list of defects. There are issues that we have because at one time these features meant survival, and now we’re stuck with them.

One very common type of confirmation bias for technical people is to believe that business people are stupid. There’s often a constant war between the technical side and the business or marketing side of an organization because each side has its own confirmation bias. If you go across that no man’s land and talk to business people, they say incredible things about technical people: that they’re negative, they never want to go along with their ideas, and so on.

It exists on both sides. It can be very clear when you sit down and talk to people about it, because you hear it in their vocabulary: “those people.”

Given that we’re stuck with these horribly retro, out-of-date brains,
how do you get people to overcome these tendencies? Is it enough to point out to a team when they’ve just exhibited a pattern of cognitive bias? Or can you bring those business people into a retrospective?

Yes, that’s the only way around it, to get rid of that us-against-them mentality. We need to say, “This business guy is one of us.” And that means that the business person has to have a trusting relationship with the team. So it should just be one person, someone who knows how to listen, who really knows how to communicate. If you have a contact like that, and he or she can bring in the business perspective without beating all the technical people over the head with it, and after a while if there is a trusting relationship that develops, then the team tends to drop that “those guys” language. After all, our guy is one of those people, and he’s okay. He’s been helping us, he’s been willing to work with us, and he understands these deadlines and our struggles.

Before I learned about the power of these biases and was able to hang around with a lot of people who do retrospectives, I believed more in the power of reflection than I do today. These biases should warn us that if we simply sit in isolation and let our own minds ruminate on what we’ve learned, whether at a personal or a team level, we’re going to get so hung up in all of the biased ways that we interpret the data we see, that we’re really not going to make much progress. Our only hope is to get together with as many diverse others as possible, who see the world differently, so that they can give us feedback on our own performance and on the performance of our team. That’s so much more valuable than just sitting alone, making a little list, and trying to act on it. It’s so hard to get past our own personal biases that opening up and getting in a room with more diverse others is our best hope.

FORREST SHULL is assistant director for empirical research in the Software Solutions Division of the Software Engineering Institute, a federally funded research and development center based at Carnegie Mellon University. He is also editor in chief of IEEE Software. Contact him at fshull@computer.org.