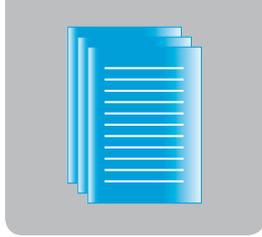




Agile Record

The Magazine for Agile Developers and Agile Testers





Agile Testing in Real Life: March 2011 – Practice, Practice, Practice!

by Lisa Crispin

There's an old joke New Yorkers tell: "A tourist asks a New Yorker, "How do I get to Carnegie Hall?" Answer: "Practice, Practice, Practice!". In Malcolm Gladwell's book *Outliers*, he discusses the 10,000 hour rule: the key to success in any field is to practice a specific activity for 10,000 hours, an idea based on a study by Anders Ericsson (http://en.wikipedia.org/wiki/Anders_Ericsson).

When and how do we, as software developers, practice? As a tester, there are many skills I need to continuously improve, from "thinking skills" such as the ability to collaborate effectively to technical skills such as the ability to design maintainable automated tests. In our daily jobs, most of us don't even think about practicing – we have real work to do. But if we never get better at our work, we'll become frustrated, we may deliver software that does not meet our quality goals, our work will be less rewarding. We must look for opportunities to practice.

In last month's column, I mentioned a few different ways testers can hone our skills, including Testing Dojos and Weekend Testing. In the past few weeks, I've had the opportunity to participate in both of these activities, as well as a Code Retreat. Here are my experiences with these learning opportunities, and the benefits I reaped from joining in. I hope these might inspire you to try one or more of these events yourself!

Weekend Testing (Actually, Weeknight Testing)

The European Weeknight Testing (<http://weekendtesting.com>) sessions are scheduled during my own lunchtime, which makes it convenient for me to join in. Each session has a facilitator and a mission. We spend an hour working on the mission, individually or in pairs, and then debrief to find out what approaches and techniques each person tried, what they found out about the mission, and what they learned.

In my first session, I paired with Darren McMillan (who's in the UK), and we decided to plan and track our testing using a mind

map. Our mission was to test TinyURL, and we tried all kinds of test cases: happy path, negative, boundary conditions, edge cases such as URLs containing accented letters, and various browsers. In the debrief, other participants told of their ideas, such as trying URLs in other character sets, turning Javascript on and off, and various heuristics that were new to me. This session gave me ideas for testing our own team's web application, which is especially helpful given that we are embarking on some major UI redesign and will need to do similar types of testing.

Testing Dojos

At Belgium Testing Days in February, I spent some time at the Testing Dojo, facilitated by Markus Gaertner. The first mission was to explore Google Refine. The first time I "test drove", we explored the idea of using Google Refine to help refactor FitNesse tests. I had trouble understanding how to use the application (which to me indicated problems with it), but we got far enough in the time period to see there might be possibilities and it could be worth pursuing later.

In the second session I attended, Markus and I paired up to plan a house using an application called "Planning Wiz". We found it awkward that the walls and objects did not seem to 'snap together', and searched the online help for clues whether there was such a feature, to no avail. (You can read more about the Testing Dojo at <http://www.shino.de/2011/02/19/belgium-testing-days-testing-dojos-report/>).

I don't get to pair a lot at my regular job, so I appreciated the chance to practice pairing at the Testing Dojo. It reminded me of the power of having two sets of eyes and two brains. Since then, I've been quicker to ask someone to come sit with me to help solve a problem or review a test design or potential bug. I'm inspired to organize a Testing Dojo for our local testing user group.

Code Retreat, Boulder

One night I was checking Twitter and noticed a tweet about a Code Retreat scheduled for Boulder, Colorado in February. I live about an hour (in good traffic) from Boulder, and I've heard so much about code retreats, I've always wanted to participate in one. I was rather scared to sign up, because these are clearly for practicing writing code, and the only code I write is test automation code. Still, I want to get better at designing maintainable automated tests, and the organizers, including Corey Haines, encouraged me to attend.

The Code Retreat consisted of six 45-minute pairing sessions in which each pair had to do TDD in the language of their choice, trying to solve the problem of Conway's Game of Life. At the end of each session, we had to delete all of our code. This kept us focused on practicing. Between each session, people shared things they had tried and what they learned. The facilitators (Corey Haines and Chad Fowler) circulated around the room, looking at everyone's code and making observations and suggestions. (You can learn more about Code Retreat at <http://www.codereetreat.com/how-it-works.html>).

I definitely embodied the "Be the Worst" pattern at this event. Out of sixty participants, I was the only tester. My best programming language is Ruby, but I'm not that fluent in it. However, I've worked with teams who have practiced TDD and pair programming for eleven years, so at least I've seen it in action; these practices were new to many of the participants. Fortunately, highly experienced programmers were willing to pair with me, and it was fun to brainstorm new ideas for solving the code design problem in each session.

I learned more in 45 minutes of pairing at Code Retreat about good code design than I could from days of reading books. For example, though I've heard the concept that tests and code must reveal intent, I had not practiced the art of naming things well and seeing the resulting benefits. I actually got to practice these things over and over, without any pressure of needing to complete some task for work.

The following week at work, I needed to update some FitNesse tests. I noticed they were full of duplication and difficult to understand, so I invested time to refactor them and leave them in better shape than I found them.

Find Opportunities for Practice

No matter where you live, you can get some testing practice with an organization such as Weekend Testing. Check your local user groups to see if there are any Testing Dojos or Code Retreats scheduled. If you don't find anything, organize one yourself – we have such a wonderful Agile community, you can find experienced facilitators willing to help and even companies willing to sponsor.

If you have trouble acting on your desire to practice your skills, find a partner with whom you can exchange learning goals. Encourage each other at least once per week through emails, questions, offering support.

10,000 hour of practice is a lot – 20 hours a week for 10 years. So, practice as much as you can at your job, too. Don't always go with the first solution you think of. Experiment with different solutions. Get someone to pair with you and focus on doing the best work you can do. This investment will pay off by enabling you to help deliver a better product, and enjoy your work more in the process.

> About the author



Lisa Crispin

is an agile testing coach and practitioner. She is the co-author, with Janet Gregory, of Agile Testing: A Practical Guide for Testers and Agile Teams (Addison-Wesley, 2009). She specializes in showing testers and agile teams how testers can add value and

how to guide development with business-facing tests. Her mission is to bring agile joy to the software testing world and testing joy to the agile development world. Lisa joined her first agile team in 2000, having enjoyed many years working as a programmer, analyst, tester, and QA director.

Since 2003, she's been a tester on a Scrum/XP team at ePlan Services, Inc. in Denver, Colorado. She frequently leads tutorials and workshops on agile testing at conferences in North America and Europe. Lisa regularly contributes articles about agile testing to publications such as Better Software Magazine, IEEE Software, and Methods and Tools. Lisa also co-authored Testing Extreme Programming (Boston: Addison-Wesley, 2002) with Tip House.