



Agile Record

The Magazine for Agile Developers and Agile Testers



October 2010

issue 4

www.agilerecord.com

free digital version

made in Germany

© Tyler Olson - Fotolia.com

What Donkeys Taught Me About Agile Development

by Lisa Crispin

If you've met me or attended one of my presentations or tutorials, you probably know about my miniature donkeys, Ernest and Chester. Driving my donkeys is my avocation, but working with them has taught me surprising skills that help me contribute to my software development team.

Trust

Ernest, our first donkey, was rescued from an abusive situation. He was half-starved and terrified of people. It took us weeks just to get near him. I'd never worked with a donkey before, but I've ridden horses all my life, and trained at the FEI levels of dressage, so I thought training a donkey would be a piece of cake. However, Ernest was a whole different animal. After long months of experimenting and working with an experienced donkey trainer, I learned the key. If a donkey trusts you, and believes that you love him, he will do whatever you ask. If not, well – forget about it, you won't be able to bribe or bully him into doing your bidding.

Donkeys have the reputation of being stubborn, but the truth is, they have a strong need for personal safety – they're looking out for Number One. Once we were driving Ernest through a field, and he stopped dead and refused to budge. Finally I got out of the cart and looked in the tall grass ahead – there was a tangle of barbed wire stretched across, invisible in the grass. Ernest somehow knew it was there, and wasn't going to let the stupid humans hurt him.

If my donkeys are afraid of something, even something that looks trivial to me such as paper bag blowing across the road, it's my responsibility to save them from it. So, when something alarms them, I get them away from the alarming object. They know I will protect them, so now they trust that anyplace I take them must be safe.

Teams revolve around trust, too. If I don't have credibility with the programmers on my team, they won't jump to fix a defect I report. Maybe they think I'm trying to get them in trouble or make myself look good. The business experts won't trust us to manage

our own workload if we never deliver on the commitments we make. If any team member sees a problem, but doesn't feel safe to raise the issue with the rest of the team, that problem won't get fixed, and that person won't be happy and productive.

It might take a long time to build up a trusting relationship – it sure did with Ernest – and it doesn't take long to destroy it if you do something harmful. But it's worth the effort. My teammates and I trust each other. If anyone needs help, they get it right away. Here's an example. Recently, one of our Canoo WebTests GUI regression scripts began to fail. A script was clicking on a link, and the resulting page returned a 404 error. However, the GUI still worked fine manually in the browser. This occurred right after a programmer checked in a refactoring. He couldn't believe his change caused the problem, but he trusts me to give him honest information – the test script had not been changed, and it had passed up to that point, so it must be something he did. After we both spent hours of research, he found that when he moved a Velocity macro into a module using IntelliJ Idea, the IDE itself made other code changes without his knowledge, trying to be "helpful". Some Javascript includes were lost, causing the 404 "behind the scenes", breaking the test. Without trust, the situation could have degenerated into a "blame game". Instead, we worked together until we solved the problem.

Our business people trust us, so if we need more time to deliver a feature the right way, they wait. We trust that the examples they give us for desired system behavior are accurate. So, we're able to deliver business value steadily and reliably.

Donkey Energy

Speaking of steady and reliable – these are two central attributes of donkeys. Ernest and Chester love to work. Chester is younger, and likes to play the clown. But hitch him to a cart or a load of hay to haul, and he focuses on his job. Donkeys don't set the world on fire, but they throw their shoulders into their work and go one step at a time. Ernest isn't flashy, but he has won the Castle Rock Donkey and Mule Show Obstacle Driving for Minis competition

five times. Chester might be a miniature donkey, only about a meter tall, but he can easily pull two adults in a cart over hill and dale and even through water or snow. As a team, they work the dressage arena every week, a tough job in the deep sand, each one pulling his weight. They never quit, so I have to be careful I don't present a challenge that is too big.

Because my donkeys know I have their best interests at heart, they're happy to try new experiences. Last year, I bought a four-wheeled buckboard wagon, much larger than the two-wheeled carts they had pulled before. The first time I hitched them to the wagon, they willingly adapted to the new situation and learned along with me. Periodically, we work with trainers to take our skills to a new level.

In my experience with agile development, slow and steady wins the race. I don't know if my team is one of those "ultra-performing" teams, but I do know we deliver significant business value to production every two weeks, and the quality of our product exceeds our customers' expectations. We don't have peaks and valleys; we focus on finishing one story at a time, and we finish several over the course of a two-week iteration. Sustainable pace rules – it allows us to continually deliver value without burdening ourselves with too much technical debt.

Like donkeys, software teams need good care and feeding: we need time to learn, time to experiment and improve our process. With a nurturing culture, we continue to do our jobs a little better every day, expanding our abilities. We can adapt to whatever curve our business throws us.

Enjoyment

Donkeys really do love to work. If they see other donkeys getting to work while they sit idle, they look dejected. They also seem to love adventure. They're always up for a road trip – they leap into the horse trailer (which is quite a big leap for these small donkeys). They might be going for a trail drive near the mountains, or going to a school (inside the school building, even) to be hugged by children. They might be going to a donkey show or for a hike – it doesn't matter, they clearly enjoy the journey.

Watching them reinforces for me how important it is to love what we do. Enjoyment is a key agile value. We must take pride in our craftsmanship, satisfied to deliver the right product to our customers, able to do so while maintaining a sustainable pace. When I first started in the software business, I thought it was something to do until I figured out what I wanted to do when I "grow up". Finally I realized I was passionate about quality and making a difference. I love being part of a business, able to contribute to its success in many ways. When every team member has this passion, and every team member is fully engaged in the process of building the best possible software, that's a joyful and productive team.

Donkey playtime reminds me how important it is to celebrate success. When work is over, Ernest and Chester play hard, chasing each other, engaging in tug-of-war, whacking each other with toy balls and feed tubs, stealing things the careless humans set

down. On my development team, we often are so heads-down in work we forget to stop and reward ourselves. It's fun to play games, enjoy treats, have a celebration, and remember why we work so hard.

What You Can Learn from Donkeys

Take a lead from Ernest and Chester. Work on building trusting relationships and nurturing a learning culture. Create an atmosphere of personal safety in your team and organization. Work steadily at a sustainable pace, keeping focus on the next goal. Anticipate adventure, and enjoy honing your craft. Celebrate when you achieve goals, big and small. You'll discover one truth about agile development: it means always finding good ways to deliver the highest quality software, satisfying your customers and yourself. ■

> 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.