How Albina Wrecked my Tester Evaluation Method

This is a story about a moment in my personal and professional development; about how one tester convinced me, by her example, that I was systematically discriminating against some good testers. After I experienced working with Albina, I stopped believing that a job interview can be a good enough method of evaluating talent.

When Albina Kuzmina (https://www.linkedin.com/in/akuzmina) interviewed for a testing job at SmartPatents, in 1998, the most striking thing I recall is that she answered *every* question with "I don't know."

Now... this can't be true. My memory *cannot* be accurate (also I recently discovered that I had misremembered her name, native language, and homeland). Still, for whatever that matters, my memory tells me she couldn't answer my questions. I would ask her "how would you test..." this or that and I recall her expressing no definitive answers and no particular confidence in her knowledge of testing. On top of that it was a struggle to understand her thick Russian accent.

I did not find her impressive except for two unusual things: *she solved a certain brainteaser*, and *she didn't get angry with me*.

(I like using brainteasers in interviews. A lot of people don't think they help, and I understand how puzzles are often used in silly or abusive ways. But they can also be used in a flexible and compassionate way. When you don't take it too seriously, it can provide a sort of neutral context for discussion about how people think. If I use a puzzle question in an interview and the candidate pushes back about how puzzle answers aren't a reasonable basis for hiring people, my response today would be "Okay, maybe you're right. What do you think would be a better basis? Let's develop a different way for you to show us what you got.")

The puzzle I used was this one: "if ABCDE x 4 = EDCBA, and each letter represents a unique decimal digit, what are the values of those letters?" I gave her the puzzle, left the room to give her a chance to work without me staring at her, and when I returned ten minutes later she had solved the puzzle. That's fast for this kind of puzzle. Even if she had simply memorized the answer, that at least tells me she's the kind of person who memorizes answers to random puzzles. And we didn't have smart phones in those days so I know she didn't just get the answer from an Alta Vista search.

I didn't fully know what to make of that, since she didn't look or sound like puzzle solver. I have since learned that puzzle solvers don't always advertise their acumen with the typical nerdy look and or technical speaking style that I, for instance, strive to cultivate.

The other thing that impressed me is that when I put her under pressure, she didn't get angry. I had put eight other candidates through the wringer, and some of them had walked out of the interview. I yelled at one guy because he said he couldn't multiply a number by four without using a calculator. Albina just serenely admitted that she could not answer my questions. She must have told me other things. All I can tell you is that none of it registered in my mind, except

“I don’t know.” (Albina, reviewing a draft of this article, reminded me that she was not very comfortable speaking English, in 1998, as well as being relatively new to testing.)

If she could calmly take that heat, I concluded she would at least not be an irritating presence on the team, and that she would probably do what she was told. Over time, I believed she would learn how to test.

Also, I was just sick of recruiting. I decided to settle for her.

Bug Whisperer

Again, my memory may be faulty, but here is what it tells me: within five days of starting in the team, being given little or no documentation and not speaking to developers, she was finding more and better bugs than the other guys who had been on the team for months or years. I was gobsmacked. I ask her how she was doing it...

"I don't know," she replied.

She hunted bugs the way a dervish spins. I would ask her how she found a bug and she would always say some variation of "I looked at it and saw it was not right." I would suggest that she analyze specifications. She would reply "Don't bother with spec. Show me product. I find bugs." I would suggest that she speak with the developers. She would reply "No, just show me product. I find bugs." Although her spoken English was broken, her written bug reports were crisp and informative.

I know now Albina had a sort of testing personality I would not name until many years later: the *empathic* tester.

An empathic tester prefers exploration and sensemaking and connecting to the customer experience. (http://www.satisfice.com/blog/archives/893) They operate more by feel than by explicit analysis. Since I'm an analytical personality who works with numbers and pictures and words, it's all too easy for me to dismiss empathic testing. It can look, to me, like random bumbling around. In my conception of testing at that time, someone like Albina should not exist. Yet there she was.

I sometimes described her as a sort of automatic bread-making machine—the kind where you give it ingredients and it does everything automatically. You don’t tell it how to make bread, and it can’t explain itself, either. She was like that with bug reports.

But this is a dangerous analogy. A bread-making machine is a robot. It follows a strict algorithm. It has no agency. It is cut off from the social world. A tester is in a weak and dangerous position when thought of as a robot.

So, I resorted to sitting with her to personally observe her methods. That worked well. I could plainly see how she worked in many cases. I could name them. *Input constraint attack… comparable products oracle… blink test… explainability oracle…* She would say "Oh, is that what it's called? I just call it finding the bug."

I had underestimated Albina. As her bug reports settled over the office like ash from a new volcano, I began to reconsider my hiring filters.

I shudder a little to think of the many people I didn’t hire because I focused on my Apollonian ideal of a tester; a tester with a silver tongue; an analytical tester. Maybe that guy who couldn’t multiply by four had something else to offer my team? I’ll never know. I didn’t care. Today, I regret my false filters and hope to live them down.

Lessons Learned

- If I were to interview someone now, I would not think I could evaluate them through conversation alone. I would give them a take-home testing challenge (which is better for introverts). I might also have them do a live testing demonstration. Even then, my preference would be to have them work on contract for a while before deciding whether to offer them a salaried position.
- When I evaluate a tester now, I prefer to think “what does this person offer?” rather than focusing on desirable things that they can’t or won’t or don’t do.
- I am now more comfortable working with a team of very different people— different from each other and different from me. I call it an elliptical team, because it’s like a stack of ellipses at different angles. None of us are well-rounded, but together we tick the boxes.
- I can’t say I have overcome my sexism, extravertism, analyticism, and general elitism. I wouldn’t believe anyone who said they did. But because of my experience with Albina, and a few others like her, I am more aware of how these things might blind me to the value certain kinds of people might bring to my team. And telling stories like this helps me get better—each telling forces a reconsideration of these lessons.

James Bach is the author of Secrets of a Buccaneer-Scholar, and co-author of Lessons Learned in Software Testing: A Context-Driven Approach. His hobby is using hyphens in book titles. He also teaches software testing around the world.

Thank you reviewers: Albina Kuzmina, Anne-Marie Charrett, Anna Royzman, Maria Kedemo, Helena Jeret-Mäe, Ajay Balamurugaras, Damian Synadinos, Griffin Jones, and Paul Seaman