Help Wanted

Hiring a Tester with an Agile Attitude

- At least three years of testing experience, preferably with Web-based application.
- At least one year of experience automating tests
- At least one year of experience with SQL and Oracle or another relational database
- Programming experience (in any language) a plus
- Good problem-solving skills
- Courage to learn, ask questions, and ask for help
- Unix administration and/or scripting skills (a plus but not required)
- Experience with open source test tools such as Fit, FitNesse or Canoo WebTest a plus
- Desire to work on an agile software development team

by Lisa Crispin

Through mailing lists and conferences, I know a lot of people I would call "agile testers." And that's just the type of tester we were looking for when my team recently began the hiring process: a professional tester who embraces change, collaborates well with both technical and business people, and understands the concept of using tests to document requirements and drive development. Agile testers generally have good technical skills, know how to team up with programmers to automate tests, and are also experienced exploratory testers.

Our original job posting briefly described our Scrum process and Extreme Programming practices, and asked for the following:

- At least three years of testing experience, preferably with Web-based applications
- At least one year of experience automating tests
- At least one year of experience with SQL and Oracle or another relational database
- Programming experience (in any language) a plus
- Good problem-solving skills
- Courage to learn, ask questions, and ask for help
- Unix administration and/or scripting skills (a plus, but not required)
- Experience with open source test tools such as Fit, FitNesse, or Canoo WebTest a plus
- Desire to work on an agile software development team

This posting elicited many responses but not very many felt right. We interviewed three candidates, but none knocked our socks off. The programmers on the team were clearly more comfortable with someone "techy," but we also wanted someone with the people skills needed to collaborate with business people and elicit examples and requirements. What did we need to do to attract the right type of candidate? What skills, characteristics, and experience should we require that might produce candidates with the right attitude and mindset? How could we better screen candidates so we don't waste time interviewing the wrong ones?

I turned to the agile testing community for help via the agile-testing Yahoo! Group mailing list. Based on input from Kay Johansen, Janet Gregory, Jonathan Kohl, and others, I put together a strategy.

One common thread in our mailing list discussion about hiring agile testers is that skills are important—but attitude counts more. As Janet Gregory put it, "Without the attitude, the skill is nothing." Kay Johansen, who has successfully hired agile testers, expanded her ideas on the characteristics of an agile mindset:

- Customer focused
- Results oriented
- Craftsmanlike approach
- Continuous learning
- Competitive
- Collaborative

SEEKING WHERE?

Our original posting was placed on just about every mailing list we could think of: local QA groups, agile groups, and local IT job lists. Naturally, I also used my professional network to try to get recommendations.

After regrouping and restarting our search a couple of months later, I put a new, improved posting only on the local QA user group jobs list. This produced a smaller pool of candidates, but they fit our requirements much better. I also continued to ping my network.

The new job posting emphasized technical skills. Here are the new and updated items we asked for:

- Experience writing black box and GUI test cases, designing tests to mitigate risks, and helping business experts define requirements
- Experience writing simple SQL queries and insert/update statements, and basic grasp of Oracle or another relational database
- At least one year of experience with scripting languages (Perl, Ruby, Python, Jython, Groovy, Tcl), and/or open source test tools (Fit, FitNesse, Canoo WebTest, HtmlUnit or others), and/or commercial test tool scripting languages, and/or programming in any language
- Knowledge of basic Unix commands; Unix shell scripting skills a plus
- Experience collaborating on some level with programmers and business experts
- Experience in context-based, exploratory, or scenario testing a plus
- Ability to work as part of a self-directed team, in which you determine your tasks on a daily basis in coordination with co-workers, rather than waiting for work to be assigned to you

Each candidate who responded to this posting seemed like a better fit than any from the first go-round. Now the question was how to sift through them to find just the right one.

SEEKING A STANDOUT

Phone screening didn't take long. If a candidate hadn't even noticed the part of the job posting describing agile development, I crossed that person off the list. Same thing, if the answer to "What do you do for professional development?" was silence. I listened for signs of "QA Police" mentality—which just doesn't work on an agile team. I wanted the candidates to have done some research, if not about our company, then about agile development. Our tester would be excited about trying agile development. Our tester would be someone who is active in local user groups, contributes to projects on

Joining an Agile Team

by Joe Yakich

During my job search, I wasn't interested in a particular process or technology as much as I was interested in finding an opportunity to work with friendly, talented folks, and perhaps even enjoy my job. When I started my new job, I didn't know agile from Adam, XP meant Windows XP, and Scrum was something that happened only on a rugby field. With all that pent-up ignorance, I naturally experienced some surprises at my new job.

Perhaps the biggest surprise to me was the pacing of the project. I was used to waterfall-type projects, both as a test engineer and as a software developer, so the flurry of task cards and test cards migrating across our Big Visible Task Chart from "To-Do" to "In Progress" to "Done" was completely new to me. In most of the projects I've worked on, there's a recovery period following a release, which is often completely necessary since you're totally exhausted from the nightmarish hours you've worked to finish the project. In our agile world, however, there's no dead time after a deadline—it's on to the next sprint. Being able to work efficiently and consistently in this environment is something I'm still trying to get better at.

Another surprise was how fun and emotionally satisfying it can be to sign up for a task, work on it, and move it to the "Done" column. There it is up on the Big Visible Task Chart—everyone can see what you're doing and how you're progressing on your current task. In some environments you may not even know how the project is going: Are we on target, are we behind schedule? Gantt and Pert charts can be incomprehensible—you're more likely to hear the unvarnished truth about the project's status at the water-cooler. But in our brand of agile, it's right there on the wall.

I was surprised to be part of the estimating process—surprised enough to hardly know how to participate! Deadlines previously had come down from Above, and in those positions my job was essentially to do as much testing as I could before the ship date. To have input into such a basic part of the process continues to delight me.

I think a few things can help one prosper in a fast-moving environment like mine. First, be prepared to do everything differently than you've done before. This can be a little scary, but it is also a great opportunity to learn new skills. Second, maintain your sense of humor. Agile environments, with their emphasis on interaction, can be a little intimidating at first (especially if you've been used to interacting solely via specifications and defect reports). Third—and most importantly—get someone to mentor you and help you learn the processes and techniques of the XP/Agile environment. My co-workers—Lisa Crispin in particular—have been invaluable in this regard. And last, have fun! Working in an agile environment will quickly make you loathe to return to the (not so-good) old days.

the side, or learns about new tools or techniques on his own. I was looking for that agile attitude!

The actual interview was more work. Some teams set up open-ended scenarios to see how the candidate would test a feature or solve a problem. This lets you see how the person thinks, and if you work through a solution together, you can see collaboration skills, too. We tried this in our first round of interviews. I gave the candidates a simple story of a feature for a retail Web site, asked how the story was valuable to end-users and to the business, and what hidden assumptions might be lurking. Nobody ever understood the question, so I think it was just poorly framed. We even thought about having the finalists download an open source test tool and write a simple test of a public Web page, but I wasn't sure that this would be reasonable. I think the idea of giving a problem or scenario to work through is a good one, but I'll need to do more work to perfect one.

Instead, I decided to try behavioral interview techniques, which were recommended by several people. When we interviewed the two standout candidates after our second job posting, I asked questions such as:

- Tell us about a time you collaborated with an end-user or product owner; what did you like about that? What didn't go so well?
- Tell us about a situation where you had a conflict with a programmer? How did you handle it?

Questions such as these let us have a more relaxed conversation with the candidate and get a feel for what it would be like to work with him. Open-ended questions let you learn positive and negative qualities. Does a candidate who experienced a difficult work situation slam a former coworker, or does he explain how he worked out the problem?

We had other questions designed to see if the tester was a good fit for our environment. We don't have a full time system administrator, so the ability to maintain one's own machine and test environment is important. My manager's favorite question to ask was "Would you rather find a lot or only a few bugs?" There's no right or wrong answer, but it can reveal a lot about the tester. See the StickyNotes for more on behavioral interviews and sample behavioral questions.

(Continued on page 37)

announced the general availability of VMware Workstation 5.5, the newest release of its desktop virtualization software. With VMware Workstation, enterprises are able to create a library of virtual machines for 32-bit and 64-bit x86 operating systems—including Linux, NetWare, Solaris x86, and Windows—that replicate production environments. These virtual machines can then be rapidly provisioned and used.

Among the major new and enhanced features in VMware Workstation 5.5 are:

- Sixty-four-bit guest support for AMD Athlon 64 processor-based systems and systems using Intel Extended Memory 64 Technology (EM64T) with Intel Virtualization Technology (experimental)
- Support for two-way Virtual SMP (experimental)
- Enhanced VMware Virtual Machine Importer
- Enhanced Command Line Interface

Virtual machines created with VMware Workstation can be deployed to VMware Player, enterprise-class VMware GSX Server, and data center-class VMware ESX Server platforms. Once deployed on GSX Server or ESX Server, VMware VirtualCenter can manage these virtual machines.

For more information, visit www.vmware.com.

Parabuild 2.0.4 Adds Integration With WinCVS

MOUNTAIN VIEW, CA—Viewtier Systems has released version 2.0.4 of its software build management server, Parabuild. Parabuild delivers uninterrupted daily builds, helping software organizations reduce risks of project failures caused by broken code base. Version 2.0.4 is a maintenance release that delivers bug fixes to the server and adds integration with WinCVS.

Key Parabuild features include the following:

- Continuous integration builds
- Daily, nightly and QA builds
- High-speed Web interface
- Two-minute installation; fifteen minutes to first build
- Integration with ClearCase, Perforce, Visual SourceSafe, Surround SCM, CVS, Subversion, and WinCVS
- Build scripting-shell scripts, Perl, make, MSBuild, Electric Cloud, nmake, ANT, nANT, Maven, Jam, and VB
- Multiple platform support-Windows, Cygwin, Linux, Solaris, HP UX, and generic Unix including Mac OS X
- Release notes from Bugzilla, Perforce jobs, and Jira

For more complete information, visit www.viewtier.com.

Mindreef Announces Web Services Lifecycle Collaboration Platform

ORLANDO, FL—Mindreef has announced Mindreef Coral, a Web services lifecycle collaboration platform for organizations building Web services and service-oriented architecture. It supplies teams with servers that can be linked together. Each server acts as a hub, housing Web service data and containing XML-aware tools that allow team members to govern, test, diagnose, and support Web services collaboratively.

For IT professionals who operate at the XML software layers, Mindreef Coral includes a powerful set of extensible tools, including foundational governance, multi-role testing, collaborative diagnostics, and lifecycle support. Mindreef Coral's tools operate on Mindreef shared workspaces (MSW). In addition to Web services artifacts like WSDL files and SOAP messages, an MSW can contain a playable script, simulation data, and descriptive notes. MSWs can be shared across teams or, using the community portal, with the Web services community at large.

For more detailed information, visit www.mindreef.com. {end}

Help Wanted

(Continued from page 18)

OUR HAPPY ENDING

We finally found the perfect person for our team. You can read his thoughts on making the switch to agile in the "Joining an Agile Team" sidebar on page 18. He has extensive experience both as a programmer and a tester, and he was eager to join an agile team. He is such a good fit, we can't imagine the team without him. In his first two weeks, he paired with a programmer to automate some tests, used exploratory testing to find bugs we'd all missed, and paired with me to creatively solve a testing dilemma we'd pondered for ages. His technical skills are an added benefit. He's super with SQL, and his Perl experience helped us plug a gap in our test automation. More importantly, he's skilled at eliciting requirements and examples from our business experts, happy to jump in on any task that needs to be done, and comes up with innovative solutions to problems facing the team. He has the all-important attitude—plus the skills.

I'm glad that we took the time and trouble to be sure of the abilities and experience we needed in a tester and to refocus our search to find that person. {end}

Since 2000, Lisa Crispin has been a tester on agile teams that develop Web-based applications. You can often find Lisa at agile- and testing-related conferences, user group meetings, and seminars in the US and Europe, helping people discover good ways for agile teams to do testing, and for testers to add value to agile teams. She co-authored Testing Extreme Programming *(Addison-Wesley, 2002) with Tip House. She contributes agile testing articles to magazines and newsletters such as* Better Software, Methods and Tools, Agile Times, *and* Novatica. *Find out more about Lisa's work (and see pictures of her miniature donkey friends) at http://lisa.crispin.home.att.net.*

Sticky Notes

For more on the following topics, go to www.StickyMinds.com/bettersoftware

- Behavioral interviews and sample behavioral questions