



Quardev Monthly

Volume 4, Issue 3

March 2010

Welcome!

Welcome to the March edition of the Quardev Monthly!

We look forward to sharing insights and helpful information in the areas where we know a thing or two - testing, quality assurance, technical writing and documentation, project management, and consulting.

In this month's article Jon Bach explores the question of whether start-ups (or any

software company with tight budgets) really need to spend time and money testing their software, and if so, how they can mitigate some of the expense by using free tools and unit testing in concert with a trusted test resource.

Enjoy the newsletter with our compliments and please contact us with questions, comments, or article ideas.

-The Quardev Crew!!

The Lament from Software Start-ups: "We Can't Afford Testing."

By Jon Bach, Manager for Corporate Intellect, Quardev, Inc.

We've heard this concern before, and rightly so. Software testing can be expensive, especially for onshore labor. Cost is on everybody's mind, especially in this economy. For smaller companies, offshore testing may be appealing for its cheap hourly rate, but it actually may be more expensive because of the time and effort needed to manage the project.

So what to do?

We tell them something that may seem counter intuitive:

"Maybe you don't need testing."

What?!?

Here's why:

- Maybe what you're shipping (deploying to the web) only has one line of code that's different and it was unit tested by the pro-

grammer and was determined to have caused no new problems.

- Maybe the app you're releasing is a beta and so bugs are fine (and expected).
- Maybe the app is even an Alpha even more buggy, but it's going to a select group of passionate early adopters.
- Maybe the app is only a demo, a mock-up for a trade show presentation. Not fully functional, but what will be shown is limited to a narrow set of features.
- Maybe the app is shareware, so expectations are low.
- Maybe the bugs you know about lie in areas that are not likely to be used, so if there are bugs,

(Continued on page 2)



Software Start-up Lament, Continued

(Continued from page 1)

your users and customers will forgive them, or not care.

If, on the other hand, your customers or stakeholders might have cause to lower their confidence in you, or see no value in what you've produced for them, or your app makes it harder (not easier) for them to use the solution you had in mind, then you may want to think about different ways you can invest in revealing meaningful problems first and fast.

If it hasn't become apparent yet with the examples above, the real issue at play here is risk – the likelihood of something bad happening and the impact to your customers when it DOES happen.

The definition of a bug is “concern about the quality of the product.”

“Quality” means “value to some person we care about.”

Therefore, think of the word “bug” as “the risk that software will not meet customer desire.”

If what they want is what they get, then no problem. The risk is if what they Desire and what they Get is so horribly different from each other that it causes a range of feelings from mild concern to frustration and anger.

We Tweeted this question to some Quardev colleagues: “*What is your advice to start-ups who say they can't afford testing?*” We gave them 140 characters to frame their advice.

Rosie: “Once small bits of testing are done they can get a better idea of the benefits. The big picture is often too scary.”

Dale: “Would you benefit from having accurate, timely, relevant information about the system? Can you afford that?”

Joe: “Do they understand what it means to test their product or service? Education is what is needed here. Show value of testing.”

Lisa: “Testing isn't a phase, it's an integral part of development. You can't really separate it out.”

Lanette: “The right question to ask is, how do I find the RIGHT testing to return value right away?” (as well as) “The only thing more expensive than testing is missing critical bugs that sink the whole company.”

Mark: “Can you afford 'not' to test; if a critical issue gets into your production that would sink your investment, is it worth it?”

Dawn: “Testing pays for itself in customers who trust you and your product.”

We also called one of our Development Partners, Matt Paulin, principal at PugetWorks.com who said that unit testing by programmers can help mitigate risk of having no budget for testers, as well as using dev-centric tools like Fire-Bug and Jprobe which can alert you to some basic problems in the code.

Given that there are at least 36 different major testing considerations (Capability, Reliability, Performance, Scalability, Usability, Portability, etc.) and that no company can execute every testing idea imaginable, it may be more effective to think about risk – likelihood of problems happening and the impact they might cause. It may just take a one-hour brainstorm with your staff to know what worries and concerns they have about the current build or proposed release.

It's a skilled tester's imagination that leads to test ideas that help you learn about the current state of the software. Good testers bring risks to light for you to discuss (and mitigate). For good reason, knowledge is power, and as our colleague Elisabeth Hendrickson says, “a test is a good response to ‘I don't know.’”