

Storytelling & Software Testing

Submitted by **Karen N. Johnson** on Fri, 07/03/2008 - 14:31.

Sometime about a year ago I became interested in storytelling. I'm not sure how it started; perhaps I stumbled across a book. I know I started explicitly seeking books on the subject last summer and by the end of the summer my fledging theory was cementing that storytelling applied to business wasn't crazy.

Can storytelling be applied to software testing? What elements of storytelling can I use? How can I weave elements of storytelling into the work that I do? Those have been guiding questions.

I also came to realize my theory wasn't unique as other people have found storytelling in business settings can work and work well. There's a significant collection of work that's been written on the subject.

I consider myself a practical person, grounded if you will. I have minimal interest in theories or topics that I can't do something with. It might be my New England upbringing or it might be that I'm a minimalist with limited patience for anything that I cannot *use*. Even when I sat in Edward Tufte's class last summer, I remember thinking well ok now all this is fascinating but how exactly am I going to apply this? I don't have lots of time to generate the graphics I need, I don't have staff and the projects I work on won't wait until I've selected a harmonizing color palette. What will I *do* with this information – me personally, concretely and specifically? How will I apply this?

So I started investigating storytelling with healthy skepticism. I majored in journalism and have years of training to be focused on the facts, focus on what can be proven. And testing software follows a factual path closer to a reporter's background than might seem after all reporting bugs is about providing evidence. Even my childhood geared me towards the factual and the analytical since it was my dad who taught me to argue based on facts and not emotions. Stories? Aren't they for children? Fables would seem a doubtful topic for someone like me in a field such as I am.

But somewhere along the way a couple of things started to happen to me with my work. I felt surrounded by facts, stacks and stacks of facts. How many builds did we have before we shipped? How many defects have been found? On and on with the facts and metrics. I think the American culture is a bit obsessed with facts too – the daily paper is loaded with useful facts and stats like the stock market and sports page. The paper is also loaded with bizarre facts that I don't know what to do with (like the first octopus with 6 legs was just found) fabulous but what am I supposed to *do* with this information? Facts roll out of my mind quickly, I don't retain them. It's like reading about the national deficit – interesting isn't it that the larger the number the less meaning it seems to have and that without meaning the information doesn't stick? I've grown restless with facts. My analytical brain has been saturated.

Another evolving situation that has happened to me is over the years as I moved through the alleged hierarchy of tester to test lead and onto manager. I've spent (too much) time away from the keyboard – away from direct software testing. (In fact, here goes both a fact and a digression at the same time – to return to the keyboard and the hands on work of software testing was a driving factor in my becoming an independent test consultant. I feared I was becoming a manager.)

I think part of me wanted to hear the stories I was missing, I missed the connection to the gory tales and details of the bugs. Away from the keyboard and dependent on other people's testing – even if they were my hires – I was given facts. How many defects, how many severe, how many closed on and on. But what I ask the testers I've hired over the years is – yes but what do you think about the software? How do you feel about the software? Do you think it's good enough to ship? I realized that I didn't want just the metrics.

One advantage in having tested for years before managing other testers was being able to ask questions like: Have you seen bugs you can't reproduce? Do you think there are issues still lurking in this product? Are there bugs you haven't reported because you know the bugs that cannot be reproduced will be disregarded? Give me your *impressions*. I often ask the testers I hire – I ask direct hires, contractors and outsourced teams that I've managed – what are your impressions. I guess what I've been asking for is, tell me a story. Tell me about your experience with the software.

This quote encapsulates what I've been looking for in a response from testers: Telling a good story is like giving a mini-documentary of what you have seen so others can see it too. This quote is from Annette Simmons book called **The Story Factor**.

Yes, that's what I want from the testers I hire. I want to hear the tales of their time with the software without turning facts to fables. And so I started doing the same when I'm the tester for my clients. I give my impressions along with the required facts. I find windows of opportunities to talk and not just handover stats. I tell what I've seen. And hands down, it's the chat not the facts that register.

Last fall at **STP Con**, I heard **James Bach** presenting. I was wired when I heard him use the term storytelling. I don't want to make any attempt to paraphrase James because I was too excited hearing him using the term to listen well. We talked about storytelling later which for me became another confirmation that storytelling and software testing can have a relationship that's practical and valuable.

Months back I also joined a **storyteller's guild**. Now I'm amused when I might end up with a week where I attend a storyteller's guild one night and OWASP the next, not exactly the same crew of folks. But it's good to learn on a variety of topics. What I've been learning about storytelling has been fascinating. And what a wonderful guild I belong to. They've been patient as I've practically interviewed each person in the guild - asking sometimes the same questions and sometimes a chat takes us down a different path. One amusing night I had my notebook in hand and asked the guild their thoughts about the categories they believe stories fall into. I confessed I'd been somewhat analyzing the craft and when I looked up from my notes I found smiling faces and one person who simply said, yeah we know. It's hard to analyze information all day and then not apply the same approach to other subject matters.

And actually as I've been analyzing storytelling I now have a mental checklist I listen for when I hear a teller. How well can they conjure up an image in my head. Does their story appeal to my senses? Is there sound? Is there color? What details do they weave in? Is it believable? How do they deliver the story? What are their presentation skills? The list is still forming in my head but I'm learning to listen and I smile when I hear certain elements. It reminds me of learning to test software, gradually moving up the scale of

awareness.

I recently had a client – for whom I did some hands on testing and when I was done, I dutifully handed in what had been requested. I turned in defects, test charters and other artifacts that had been requested. But I also provided my impressions. I wrote a letter, not a long letter but a letter outside of my test materials that stated what I thought, what I'd experienced with the software. Stories don't take place of facts but a story can certainly give light that facts don't.

Alright, here's my WIFFM checkpoint. (I claim to provide a WIFFM in each post to help prevent me from writing a blog that lacks practical application to software testing. Sometimes I call out the WIFFM directly other times it's my own sanity checkpoint.)

If you're interested in learning how to apply storytelling here's a couple of concrete take aways. Look for opportunities to tell. When can you tell more than just the facts? And to what extent will your audience want those details. Which leads to the age-old question - who is your audience? I'd advise to factor in your current street credentials (a reference back to [Mike Kelly's post](#) awhile back about credit ratings.) It's best not to forget where you stand with a crowd before you deliver your goods.

Storytellers practice. They practice telling whenever they can; they ask people they work with and live with to listen to them. They find places to practice, they create guilds to practice. (I practice at my local guild and ask for feedback which has been valuable.)

I've questioned how they practice and there's been a range of responses. Some practice while driving, some practice at home. Some tellers record their stories, playback and listen to critique themselves. Regardless of how, every teller I've met says practice is critical.

I've heard many styles of storytelling and some I can relate to and appreciate more than others. You have to find your own style – it's true with presenting and it's true with storytelling. If you present or attempt storytelling in a style that doesn't suit you, it will ring false. Finding your style and developing your style is a topic of its own.

I've learned, not surprisingly that some parts of telling need to be planned, packaged carefully and then rehearsed before delivery. Storytelling is like presenting, many of the best presenters practice. It can often be a case of someone appearing so natural at what they're doing, you can't envision them practicing, it's become natural to them. But it takes work to make flowing stories come out with ease. And as expected the more difficult the subject matter, the more practice is important. So in a business setting if the information I'm going to share is controversial, I choose my words in advance. While I might not deliver an entire presentation of memorized information, some core elements I will memorize. I've been surprised to hear how often storytellers tell me they memorize material but then I've come to realize it's their skill in delivering that keeps me from detecting what's been memorized.

In April, I'll presenting at the [Software Test and Performance Conference](#) detailing how I've been using storytelling and what some of the tellers have been telling me. I've been reading up on the craft, attending guild meetings, finding opportunities to apply storytelling to software testing. I've been writing notes and I've built a mind map on the subject. Sometime over the next few weeks, I'll add materials to my website to share.

Karen is also presenting at **STANZ** in Sydney, August 27-28, and Wellington August 24-25, 2009 giving an updated keynote presentation on Storytelling and testing.

First published www.karennjohnson.com/blog
(www.testingreflections.com/node/view/6700)