

# How did you find that bug? Are we sitting comfortably, then I'll begin.

29 March 2018 · agile, exploratory testing, investigation, map, pre-scripted testing

---

How did you find that bug? - They asked with a sort of puzzled “he dun’t thunk like uz” look on their faces. An expression that suggested they were unsure whether to commend the discovery or gather their pitchforks and organise a well overdue witch burning.

Likewise, I now knew why they needed me. The team members were genuinely hard working people trying to build something new and exciting. But they lacked one thing, someone exploring & asking questions - trying to find out new things about their application. Exploring is literally a step into the unknown, and that can be uncomfortable for those not experienced in how to do it well.

So how did I find that bug? It’s easy to tell a story of how I tried that particular input value because... Paragraph 3 of v4.6 of the requirements document stated that the user shall indeed on occasion X given input Y in Chrome v62 do... Or spout some other overly verbose explanation of why that broken ‘scenario’ came to be valid.

But I wouldn’t be answering their question, and I wouldn’t be telling the truth. The truth of how I (and other testers) do this is harder to remember. But having a better idea of how you got there is going to help you get there again.

Let’s take a trivial example. A few years ago, working on a new editorial system for a financial news organisation: I discovered that entering a double quote character into the content search feature caused it to crash. Even worse it returned no results (But it didn’t fail to provide the users with a nasty technical error message). This had not been noticed before, yet many tests had been written and run. They had more acceptance tests than they could run in a night.

I had not examined requirements documentation for that feature. I had not perused the Acceptance Criteria for the Search feature. I had not reviewed the unreliable test automation. What I did was speak to the team and then type some quoted text into the search form. I had thought that the search should behave like Google, whereby if you include text in quotes - it will search for an exact match to that string of characters. I was interested to see what happened.

I did not know whether the search should support this sort of exact phrase matching. But I was exploring. I knew ‘searches’ [in my experience] worked like this - so I decided to check if this one did. This was the first step in a series of related tests that allowed me and my colleagues to uncover many more issues with the search feature.

When a team is focused on short-term delivery goals they can lose sight of the features not listed right in front of them. You need someone to question, someone to explore someone to point out that crashing when an editor uses a quote is sub-optimal. That's what I do at Investigating Software.