

Heurism

20 December 2010 · exploratory testing, heurism, heuristics, learning

I'm watching my son, a toddler, at play. He picks up his toy train, a hefty piece of wind-up fisher-price-esque technology, and hurls it at the water bottle. I'll not pass judgement - but suffice to say - the bottle is still standing - several other objects in the room are not. He reaches down with both arms and picks up the train again. He steps a bit further away, turns his back on the bottle, and slings it back over his shoulder. A few more similar attempts end in much the same result, Until finally the killer-move is identified: You stand point-blank over the bottle and drop/throw the train down onto the bottle.

A chip off the old block. I'm glad my son is having fun. But I'm interested - What's he thinking? No, that's not it... How is he thinking? What he's doing has strong parallels with what his father does for a living. I spend much of my time learning how [for example] a tool works or, maybe more often, how they don't work. If that takes the application of a 'surprise' heavy load well that just adds to the fun.

I think back to my recent reading and Rapid Software Testing course. If there's one word that sticks in my mind from both it's: 'Heuristic'. So is he applying heuristics? and if so which ones? He needs to know if he's hit the bottle with the train - that's more of a matter of observation. He clearly has a hypothesis that lobbing the train can 'take-down' that evian bottle (Which he generally disproves - clearly a fan of [Popper](http://en.wikipedia.org/wiki/Karl_Popper) (http://en.wikipedia.org/wiki/Karl_Popper)) - but again thats not really a heuristic.

It's simpler than that. I looked up Heuristic in the dictionary and didn't find much clarification. However, I did notice the word: Heurism. "The educational practice or principle of training pupils to discover things for themselves." - [Oxford English Dictionary](http://www.oed.com/) (<http://www.oed.com/>). My son was learning for himself, he was gaining evidence for himself, empirically. He was building-up his first-hand experience. That's what I do! I can't give you much information until I actually use your system. Let me hurl a few keystrokes at the command line, and I'll give you back some real fact based information. Let me do it a few times, and I'll give you back even more information.

In fact that reminded me of a real technique I use when testing: Bug-compression. If I find a bug - lets say a user-path that has many commands or clicks, I spend a little time 'compressing' the steps - reducing the number of operations or clicks too just a few. This can be valuable in demonstrating that users could reach this 'failure state' more easily than it might at first appear. E.g.: The 'unlikely event' is now only three key presses away from the happy path.

This makes sense, My son's building the store of knowledge that firstly: he can use to generate his own heuristics and secondly: maybe apply the heuristics he's already learned. As testers we need to get into one of those feedback loops, where our actions are giving us more information that helps us find out even more. Get your hands & eyes on the controls, the logs, the database. There's a heuristic right there, built from hands-on experience: "Check the log files for error messages." It's fallible, but none the less, it's been a good way to find oddness and bugs so far.