

Why you might need testers

11 April 2017 · agile, investigation, pre-scripted testing

I remember teaching my son to ride his bike. No, Strike that, Helping him to learn to ride his bike. It's that way round - if we are honest - he was changing his brain so it could adapt to the mechanism and behaviour of the bike. I was just holding the bike, pushing and showering him with praise and tips.



(https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiGy5EsGcQMb6DS4tjr-Bg4W_TA4axKz8XvdhYUbf7vetCXzANAzxVxsub-JMjsyGd4_2aZdBGmzOS18fosCVFyPYGqyBPIld6SDMUycadbh13Clgdmali4wslpZ8Ujx02AD0lxQkewKQurw/s1600/640px-Left_side_of_Flying_Pigeon.jpg)

By [] from Peking, People's Republic of China (Down the Hutong)

If he fell, I didn't and couldn't change the way he was riding the bike. I suggested things, rubbed his sore knee and pointed out that he had just cycled more in that last attempt - than he had ever managed before - Son this is working, you're getting it.

I had help of course, Gravity being one. When he lost balance, it hurt. Not a lot, but enough for his brain to get the feedback it needed to rewire a few neurons. If the mistakes were

subtler, advice might help – try going faster – that will make the bike less wobbly. The excitement of going faster and better helped rewire a few more neurons.

When we have this sort of immediate feedback we learn quicker, we improve our game. When the feedback is about problems we don't even know might exist, that expands our knowledge letting us adapt before its painful to do so.

My son had his own plan, skill and will. He could have learned on his own. It probably would have taken longer and maybe been more painful or even dangerous. Working as a team, we achieved something, more than we could alone.

In software development that's how successful and sustainable teams work. We don't have gravity, so we need other sources of feedback. Some of those might be 'automated tests', but much like my sons own 'plan' they only covered predicted behaviour. "So I just peddle and avoid the trees - Dad, it won't be hard". We need another mind and experience to help us see outside our view of the problems. "Stay off the grass as well, it's harder and more unstable there".

That's where skilled testers come in. Some of us are highly technical and will highlight technical issues or opportunities you haven't noticed. Some of us are skilled at seeing the app from the customers view point and highlighting problems that will cost you customers. Many of us do the above and much more. This could be letting you know if we are breaking the law. We often write tools that enable us to test more things – better. Testers can be useful for just pointing out the simple stuff like – we are developing for Red Hat Linux, but 60% of our users are on Windows, and 25% are on Apple Macs.

"Our analytics will tell us what the customers are doing! We can tell if a feature or bug matters from the metrics!" Great idea. Do the analytics work well enough? Do you know if a user experiences an error? Do your analytics double-report or under-report? A second pair of eyes will probably find the analytics probably don't work as well as we'd hoped.