

Now that's what I call a phone bill, innit

I thought I would carry on in the same vein as last month and say a little more about defects in programs. In very, very crude terms, there are two kinds of defect: those you notice and those you don't. In general, people tend to get most exercised about those we notice because the ones we don't notice can't really be very important because we don't notice them. Wrong.

This all came to mind when a gentleman from Malaysia received a telephone bill for 218 trillion dollars in April. Yes, that's 2.18×10^{14} , a very big number. This was widely reported last month so I will say no more about it but the point I would like to make is that he was lucky. It is obviously wrong - even a teenager with a mobile whatsit couldn't rack that kind of money up unless they logged onto a website in the Andromeda Nebula for a year or two. Eventually even the Malaysian telephone authority responsible admitted it was an error.

Suppose however that they had falsely issued a bill for 2,180, or even 21,800 dollars? How long do you think it would have taken them even to agree it was an error? These are the defects which are much more insidious because they are not obviously wrong.

You might not think that we make such mistakes very often but you would be surprised. Numerical mistakes at the level where erroneous results seem 'reasonable' are surprisingly common. Let me take you back to an experiment I was involved with in the 1990s. It measured how accurately seismic surveying software predicted where to drill for oil. At the time, there were 9 different packages, all written to the same (mathematically defined) requirements in the same programming language in deadly competition so there was no collusion. The packages had each been in use for years and had racked up thousands of execution years figuring out where to drill an oil well. Drilling oil wells in the North Sea costs around 20-30 million dollars so its fairly important to get it right.

We decided to give them the same data and the same disposable parameters to see if they came up with the same answer, an expensive experiment graciously funded by Enterprise Oil. The slightly embarrassing bottom line was that we obtained nine different answers, but, and this is the important bit, each answer looked reasonable on its own. We then asked geologists to study these nine different results to answer the question "Shall I drill here?". We basically got a random answer, (3 yes, 3 maybe and 3 no). The variations were entirely caused by previously unnoticed software defects and this in an industry with extensive quality control processes for its software development. Furthermore, the software defects which caused the problem had been in their respective packages for between 1000 and 2000 execution years so you can't trust a program just because its been around for a while.

Of course, you might believe that perhaps this sort of thing only occurs to programmers in the oil industry but just in case, it does no harm to check the odd financial calculation every now and then just to make sure that nobody has screwed up. You may be surprised with what you will find.

l.hatton@kingston.ac.uk