So who says development is easy ?

Although I frequently and I feel justifiably complain about the quality of some software systems, I suppose I ought to represent the side of the hard-pressed developer from time to time.

The average user will by now be used to the continual need for patches both to fill in security holes and to repair a basic program function which does not behave correctly. The average user will also be familiar with how long this takes in download time, so let me take you through roughly what is necessary to fix a defect on a realistic piece of software. First of all, you have to find it from whatever evidence is presented. Defect evidence is legendarily terse if not downright apocryphal - “it doesn't work” often used to figure in support databases in my previous companies. When you have finally figured out what this means and translated it mentally into some kind of action plan, some considerable amount of time can have passed. I published a paper some years ago on C++ systems where 5% of all defects took longer than 100 hours to find and fix and 2% took longer than 200 hours. Even worse, some defects simply can't be fixed because their cause cannot be found - there simply isn't enough information. In fact, there is some evidence that this may be as much as half of all defects in some systems and there is also evidence of very long-standing defects even in critical commercial systems such as avionics software.

Let's now jump into the haystack. Speaking of which, finding the eponymous needle is simple in comparison. If you have a rummage in the System and System32 directories of Windows XP for example, you will find it comprises some 700Mb of object code. A reasonably good estimate to turn this into lines of code is 15 bytes per line meaning that your haystack contains around 47 million lines of code. Computer scientists like to talk of tokens, being a syntactic unit like a word in a book. On average there are around 5 tokens per line so we are looking at a haystack of some 250 million tokens. By comparison, this article contains around 500 such tokens so the haystack we are jumping into would correspond to a book of some 62,500 pages.

Once we have found the wrong word, sentence, page, section or possibly even chapter, we have to rewrite it to be consistent with whatever evidence we were presented with originally as well as the rest of the book. Now it gets even more interesting. In 1984 in a famous paper, Ed Adams of IBM found that for every 7 defects fixed, a new defect was accidentally introduced, (the so-called 'job security defect'). I have personally seen legacy systems where this was one to one - fix one, break something else. We have tried to combat this in many ways over the years using various grandly named technologies but we have basically failed. It remains astonishingly difficult to fix defects in a large system without breaking something else. Oh yes, and developers have to do all this often to punishingly restrictive deadlines with usually risible test budgets and as I noted in my last article, often with inadequate training.

Have I tempted you into becoming a software developer yet ?

lesh@leshatton.org