

## Forensic Software Engineering and not before time

I'm writing this article in August because I am jetting off to a conference in Melbourne on, of all things, software testing. Given the parlous state of modern software, it would be nice if more people attended conferences like this but small steps first.

There has been much discussion in the press in recent months on this topic with initiatives firing up all over the place, the Jericho Forum here in the UK, the Global Council of CSOs in the US, the US department of Energy using its purchasing clout to persuade Oracle to harden 9i, the warnings about the use of Internet Explorer and so on. Well its very nice to hear that people are beginning to worry but what may I ask took so long ? The quality of shipped software has on average been so poor for so long we have no real conception of what its like to have a software product which is delivered on time, transparently obvious to use, does not use English normally found in computer fairs and simply works first time, every time.

If you are irritated with the quality of your software, this is the bottom line: many and perhaps most of the failures we experience with software based systems could have been avoided using techniques we already know how to do. Do not accept excuses about, "we need to upgrade ...", or "the users didn't read the manual ...", (which appears to be in Klingon anyway), or "we are using old technology ..." or whatever this week's excuse is. The brutal truth is that we do not train software developers to be engineers, (so we shouldn't call them engineers), software testers frequently enjoy the same status in a company as the office cat, requirements are often missing, optional or treated as entirely superfluous, deadlines are set by pins in calendars and project planning and tracking is something that other people do. I expect some readers might be a little shocked by this but its a fairly succinct summary of the conclusions of recent reports around the world.

Even when we have buckets of money, we seem to find it hard going. Consider the F/22 Raptor, the latest and greatest fighter in the world, (they weren't going to call it the F/22 Hamster were they). According to the Washington Post in 2003, test pilots were spending 14 minutes a flight rebooting critical systems but this is now down to 'only' 36 seconds a flight. Well that's a relief. We are talking about missile control systems and so on here by the way.

So given that we are talking about it, are we going to act ? Well, every little helps. There's a new centre for Forensic Software Engineering starting at the University of Kingston this month and existing groups at Middlesex (specialising in project failures) and also Glasgow. The theme is very simple: find out what failed and how to avoid it so it doesn't happen again. When a bridge fails, we go to great lengths to find out why and disseminate this information. When software fails, we utter an oath and reboot, expunging all evidence. According to the Royal Academy of Engineering, several billion pounds a year is riding on this in the UK alone so these centres will need data and will need financial support. Your country's software needs you.

lesh@leshatton.org