

Editorial

John Mitchell

The cleaning instructions for one of my early Philishave electric razors said that I should clean the cutting heads by holding them under a running tap. I did so and simply ended up with a load of gunk in the heads which decreased the shaving effectiveness by one hundred percent. Likewise, when Alan Titmarsh said that I could turn my tree leaves into excellent compost by putting them into large plastic bags, punching some air holes and adding Gorotta, I fell for that too. All I had the following Spring was 62 large bags of smelly and slimy leaves. As you can see I used to follow instructions without too much thought on the basis that the experts knew what they were talking about. Now that I am older and wiser I try to evaluate the instructions before going through with the process to ensure that I am likely to be happy with the outcome and that there is a back-out and recovery capability. This is fine if they are in a format where you can review the entire process, but in many cases you are drip fed bits of information and have to blindly follow the path presented to you one step at a time. It's only when you have stepped off the cliff that you have a moment to reflect that the last step was not a good one to have taken. No where is this more evident than the vexed task of software installation.

Windows XP provides an opportunity to take a restore point before changing anything, which I do always, but several times upon trying to roll back it has greeted me with bad news that it is unable to restore to that particular point. Why this is the case I do not know, it is just one of those bad things in life, like wasps and flies. I have tried backing up the entire partition, but often cannot do so because the partition that contains the software for the back-up is the very partition that I want to back-up and the software states that I am not allowed to back-up a partition that is in use! Just how did they test this software before releasing it? Not very adequately, that's for sure.

Very few of my clients have a decent testing strategy or quality assurance process and even less have mechanisms in place to prevent the insertion of malicious code when an authorised change is being made. Yet all of these requirements can be built into a decent change process with a little thought. Often it is pressure from the user for speed of release which gets in the way of the quality process, but this is just a question of managing expectations by building-in sufficient time for testing and quality assurance when discussing the change with the user at the requirements stage. A thirty to fifty percent overhead is the norm, to be added to the time that the development staff require to change the code. I once ran an independent testing team and always surprised people by insisting on involvement at the requirements stage. My argument being that if I couldn't understand the requirement sufficiently enough to work out how to test it, then the development staff wouldn't understand it anyway. What was the point of waiting until almost go-live to find out that the functionality was not what the user wanted anyway? Oh, they used to fight a bit, especially the users, over my

requirement for a clear specification, but I persevered and showed them my quality triangle¹ until they were sick of it and gave in. The result was a more expensive solution than that being proposed by the developers, simply because of the thirty to fifty percent testing overhead, but they always received a quality product which gave them no problems when it entered production. To test the quality of your organisation's development process review the incidents reported to the help desk, or the availability statistics of the SLA. High incidents and/or low availability after a software release is a consequence of a poor change delivery process. Operations are on the receiving end, but the root cause is in the development section.

Being eagle eyed auditors you will have noted that it is our 40th anniversary this year which we intend to celebrate it by running a free one day event for our (paid-up) members to which we will also invite previous members of our management committee. This should be quite a good day.

In this edition you will find a paper on high trust system integrity controls which deals with a number of important issues, a summary of a report by the Audit Commission on its recent survey on computer related crime in the UK, a column from our chairman Alex Brewer, a down-under column from Bob Ashton and an update on our parent body from Colin Thompson. Enjoy!

¹ The relationship between functionality, time & cost