In my experience, some programmers and program development managers resist investing time in software quality assurance (SQA). In a recent research article on “Resistance Factors in the Implementation of Software Process Improvement Project in Malaysia,”<http://www.scipub.org/fulltext/jcs/jcs43211-219.pdf> from the *Journal of Computer Science* 4(3):211-219 (2008), the authors summarized extensive published research on why people resist SQA. Experts have found that there are several categories of stumbling blocks to integrating SQA into the software development process (Table 1, p 213):

- Human: failure to gain top-level, thoroughgoing support for process improvement;
- Political: perceptions of loss of power;
- Cultural: organizational resistance to changes in long-established patterns;
- Goals: unclear, undefined, unmeasured goals leave people confused and uncooperative;
- Change Management: SQA must be integrated with and support the mission-critical goals of the organization.

An essential step in implementing new SQA processes – and continuous process improvement (CPI) in general – in any organization thus involves convincing all involved stakeholders (employees, managers, shareholders and even customers) that the project is worth the effort. I have some ideas from teaching that may be helpful in this task.

One of the key steps in teaching is to show students why a subject is worth learning. My practice, developed through four decades of teaching, is to start every lecture with an informal overview of how a topic relates to the real world. Thus in discussing SQA in a management of information assurance (IA) course or a systems engineering course, showing students some cases where SQA was lacking is an entertaining way of bringing the message home vividly.

The Forum on Risks to the Public in Computers and Related Systems (“The Risks Forum”)<http://catless.ncl.ac.uk/Risks> of the Association for Computing Machinery (ACM<http://www.acm.org/>)<http://www.acm.org/>, ably run for over 20 years by Dr Peter G. Neumann<http://www.csl.sri.com/users/neumann>, is a goldmine of reports on the consequences – some of them hilarious – of poor software design and failures of SQA. My now-slightly-elderly supplementary lecture<http://www.mekabay.com/courses/academic/norwich/is342/lectures/csh5_ch39_software_devt_qa_supplement.pptx> from the IS342 Management of IA course<http://www.mekabay.com/courses/academic/norwich/is342/index.htm> at Norwich University<http://www.norwich.edu> has lots of slides you can use freely in your own presentation on SQA failures. Here are some of the stories that usually get my students’ attention:

- A 3-year-old gets an IRS refund for $219,495;
- Microsoft publishes an unverified Spanish thesaurus which includes insulting slurs, resulting in a public relations debacle;
- The ENT Federal Credit Union ignores months of customer complaints about their automated teller machines, allowing the defective programming to count only the first withdrawal by a customer – and resulting in $1.2M in losses;
• A dentist receives 16,000 identical copies of a tax form;
• *Flintstones* cartoon viewers in Springfield, Missouri are unexpectedly switched to watching the Playboy Channel;
• A vagrant applies to Sandoz for a $2 refund on a used bottle of Ex-Lax but receives a check in the amount of his ZIP code – $98,002 – and promptly disappears after cashing the check;
• A programming error in the First National Bank of Chicago system adds ~$900M (yep, million) to each of 900 customer accounts for a total accounting error of ~$764B (yep, billion);
• Smith Barney adds $19M to each of 525,000 accounts (for only a few minutes) for the largest accounting error in history: ~$10T;
• Los Angeles County underpays its employee pension fund for 20 years due to a programming error, resulting in $1.2B in unexpected liability;
• and my favorite demonstration that nobody can do mental arithmetic anymore – a secretary accuses a professor of creating 4,294,967,026 copies in two weeks (~3551 copies/second continuously 24 hours a day) because the photocopier says so – and removes his photocopying privileges!

Next time, I’ll present an interesting study of the value of automated SQA testing tools.

* * *


Copyright © 2010 M. E. Kabay. All rights reserved.

Permission is hereby granted to Network World to distribute this article at will, to post it without limit on any Web site, and to republish it in any way they see fit.