

The MegaPole

Carl A. Singer, Ph.D.

If you think losing \$400 million in 45 minutes can't happen to you.

Many years ago a backhoe severed a major long distance trunk causing a temporary outage and calling into question the pros and cons of laying cable alongside a highway. Not to be outdone a software “typo” which was promulgated throughout the network (switch-to-switch-to-switch) also quickly and painlessly brought long distance to its knees.

I was Manager of Quality and Cost Initiatives at Bellcore at the time. (Bellcore was the piece of the former Bell Labs that was owned by the baby bells.) In an attempt to communicate this phenomenon I equated the software glitch with a hurricane or tornado knocking down telephone poles – hence “*megapole*.” How many thousands of poles would need to come tumbling down to cause the same damage. {It should be noted that the term “mega-hoe” was inappropriate.}

Much of our experience with building software seems to have its roots in managing the building of hardware – physical things. But the analogy soon breaks down. Even the novice home builder knows that the foundation must precede the roof and that the plans must be in place prior to starting. The builder knows that upon completion a building inspector will look at the finished product and determine if the home is ready for occupancy.

Unthinkingly, we carry this analogy into the world of software. Are the outlets properly grounded? Are the beams of sufficient strength and spacing to support the weight? What outlets? What beams? There are many who confuse Quality Assurance with Testing. This morning I saw a job advertisement from one of the few remaining money center banks advertising for a Senior Vice President Quality Assurance – the job description was for someone to lead testing. Preaching to the choir – you cannot test quality into software!

So what is one to do? – the answer: Process. One can examine process – both the soundness of the processes and the organization and projects’ adherence to those processes. Processes cover all domains of the software lifecycle including requirements, design, coding, configuration management and (yes, even) testing. Organizations document and share process – thriving organizations hone process by looking at “best practices” and sharing successful processes.

Traditionally process has been associated with work breakdowns structures, nested series of tasks. A more robust method of capturing and promulgating process is described in my IBM Systems Journal Article: *Context-Specific Intellectual Capital – the Next Link in the Knowledge Chain* (Vol 42, No 3, 2003.)

Regardless of how you create, capture, document, share, enforce, test process it must be the core of your software development efforts. Process Makes Perfect.

Dr. Singer has extensive software development, quality assurance and project management experience. His certifications include PMP, ISO 9000 Lead Auditor, TickIT auditor. He currently serves as Chairman of the Management Board of the IEEE Software and Systems Engineering Standards Committee. His website, www.ProcessMakesPerfect.net contains several interesting articles.