

sions that are different from the ones you think they should reach.

*Victor Skowronski*  
victor31@ieee.org

*Neville Holmes responds:*

When people think for themselves, and when that thinking is rational, the ideal democracy becomes

approachable. The other way leads to dictatorship and ochlocracy, which is no way to solve the world's problems.

### PROVENANCE AND REPRODUCIBILITY

I had the pleasure of meeting David De Roure at a meeting in 2009 where we discussed my ideas

on provenance and reproducibility as they relate to computational science and engineering. Thus, I was delighted to read his column contribution titled "e-Science and the Web" in *Computer's* May 2010 issue (Web Technologies, pp. 90-93), where he echoed my discussion of the distinction between repeatability and reproducibility. Although his column includes URLs to various websites, it omitted a URL for my work. Interested readers can find the original discussion of this topic in my paper written in 1998 and available at [www.toolsmiths.com/docs/CT199801.pdf](http://www.toolsmiths.com/docs/CT199801.pdf).

Any discussion of provenance and reproducibility for computational science and engineering that does not also address citation and attribution leads to a contradiction in terms. It is not possible to maintain standards for scholarly peer-reviewed reproducible science without proper citation and attribution. Even in a one-page editorial, care should be taken to attribute ideas and contributions to their original authors. In a four-page column, confusion quickly arises as to whether the text represents a contribution of the column's author or discussion of another author's work.

Unfortunately, failure to cite and attribute properly has not been limited to editorials or columns but has become a growing problem impacting many peer-reviewed papers. Another worrisome trend has been the growing frequency of the appearance of statements about funding that is featured prominently in abstracts and introductions rather than in footnotes or acknowledgments. But funding of any amount is not the substance of the science or engineering that should be presented in a paper for scholarly review and publication.

Additional commentary with suggestions for practices that might alleviate some of these problems can be found in a blog article at [www.portaldoors.org/Blog/PostID/5.aspx](http://www.portaldoors.org/Blog/PostID/5.aspx).

*Carl Taswell*  
ctaswell@computer.org

## REMEMBERING WATTS HUMPHREY

The worldwide software engineering community lost one of its brightest lights when Watts Humphrey, who was called the father of software quality, died on October 28. I feel the loss not only professionally, but also personally because I worked with Watts for 23 years at the Carnegie Mellon Software Engineering Institute (SEI).

I met Watts when I joined the SEI in February 1987. Watts had started a few months before I did and was just beginning to staff the Software Process Program and establish the program's technical agenda. With Watts leading the way, the members of this group produced the initial appraisal methods and evolved the five-level maturity framework, creating the Software CMM. By 1992, these concepts were becoming well established in the community, and Watts began to work on applying his theories to software engineering practice.

Watts first tested his theories on a process that he developed for managing his personal checking account. Next, he tested this on the personal software development process by writing more than 60 small programs in Pascal and C++. Then he began working with organizations to pilot this new personal process for software engineers.

Not long after, Watts published his first Personal Software Process (PSP) book, *A Discipline for Software Engineering*, and developed a course for software engineers. When I enrolled in the first PSP course offered at Carnegie Mellon University, it changed my career.

When you learn how to properly measure your own performance and analyze the result in order to improve, you get real, lasting, behavioral change that leads to performance gains and improvement. In the class, we went from underestimating our work by about 40 percent to being within a few percent under or over estimate on each assignment. By the end of the course, we had a 10 times reduction

in the number of defects that escaped to the unit-testing phase. These results were unbelievable. If I hadn't been there, I wouldn't have thought this possible. The next challenge was how to achieve similar results on real projects.

After the course, I began working with Watts to transition Team Software Process (TSP) and PSP into software engineering practice. During the course of our work together, we became close friends.

What will stick with me? First, the belief that with both the maturity model and PSP/TSP, Watts created a framework that is the right stuff for software engineering and probably most kinds of related work. It works. Second is the value of data. When you have personal data for a process, you have a fundamentally different kind of understanding of how that process works.

I will never forget the Watts-isms. Watts was a master at reducing the complex to the simple, and there are hundreds of these little gems. For example:

- *Watts on planning:* If you can't make accurate plans, plan often.
- *Watts on producing quality work:* If you want a quality product out of test, you must put a quality product into test.
- *Watts on life:* In life, we all reach the same end, so we need to concentrate on the trip. Devote yourself to excellence, and you just might achieve it. That would be worth the trip.

Watts made a commitment to improve the profession of software engineering, and when he made a commitment, he meant it. He inspired many professionals to join him on this mission, where he was equally committed to us. His commitment is now ours.

—*Jim Over, Senior Member,  
Technical Staff, Software  
Engineering Institute*