

## *Book Review*

# Facts and Fallacies of Software Engineering

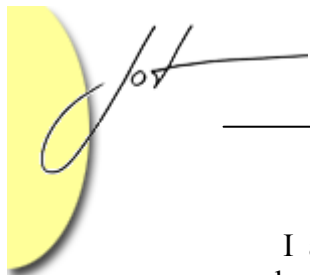
by Robert L. Glass, Addison-Wesley, Boston, MA, 2003. 195 pp., \$29.99(paper). ISBN 0-321-11742-5.

**Reviewed by Charles Ashbacher**

This book is a breath of fresh air in a world of hype, herd mentality and a continuous search for the computational equivalent of a magic wand. The players in the great game of computing are still in the process of developing the set of processes that can be followed to guarantee the creation of a successful software product. Since the field is still so new and changing at a speed unmatched in any other area, there is a great deal of room for errors and misconceptions. Glass has developed a set of what he considers the most common errors, some of which will raise your eyebrows all the way to the point at the top of your head. Conventional wisdom is often more convention than wisdom, and since his opinions are often right, all people involved in the software development process should read this book and not dismiss any of his views until you have thought them through.

There were many of the points that I found myself mentally agreeing with in a very strong way. One of the most significant is when he points out that the percentage of code reuse tends to peak in the range of 20 to 25 percent for software with general use. Which is far less than the strong advocates of reuse claim. My experience is completely consistent with the claims put forward by Glass. In circumstances where the data and goals had changed very little since the software was originally written, we still found it more cost effective to rewrite much of the software rather than reuse it.

His first fallacy, “You can’t manage what you can’t measure”, is clearly very controversial and one that I am in complete agreement with. A great deal of money has been spent on tools to measure the progress of a software project, which means that many people, sellers and users alike, have a large stake in believing that measurement is a complete solution. However, there is strong evidence that every metric used to measure software development fails in many circumstances and there is overwhelming evidence that many groups simply ignore the measurements and suffer no ill effects.



I am also in agreement with Glass's opinion that the overwhelming majority of productivity tools are briefly tried and then gather an impressive layer of dust on a shelf. The cry over lack of appropriate tools is one that I have heard and is often voiced by people who need an excuse for poor performance. Good developers really need only a few tools, the most important of which is an interactive debugger. His point that the last thing the IT field needs is another methodology is another that is very welcome. There is an excess of methodologies and in my opinion, we really do not know how to effectively apply the ones that currently exist. Therefore, adding more is certainly not the answer.

Sometimes the most substantial changes are triggered by those who only state the points for debate. Martin Luther nailed his 95 theses to a door and set in motion a movement that changed the practice of religion forever. Glass has put down 55 facts and 10 fallacies that also challenge the established order of software development. Of course the consequences of his actions will not be so profound, but they are worth reading and he is doing the IT field a great service by raising his voice in opposition.