Walter Shewhart and the Philosophical Foundations of Software Engineering

David Grier*[†1,2]

[1]IEEE – United States
[2]George Washington University – United States

Abstract

To create engineering standards of practice for software, the early practitioners of this discipline had to accomplish two goals. First, they had to materialize an immaterial artifact, the program. They had to conceptualize software in a way that could be measured. Second, they had to introduce the concept of randomness into a decidedly deterministic framework. Practically, these two goals forced them to reject two dominant modes of engineering practice, those of mechanical and electrical engineering. Historically, this was a relatively straightforward task as mechanical and electrical engineers proved easy to ignore. They wanted to dictate the nature of the new engineering field but were unwilling to do the work to shape it.

The early software moved quickly to base their practice on the works of Walter Shewhart(1891-1967), who was one of the founders of industrial engineering and quality control. In particular, the work of provided the foundation for the 10 IEEE standards that formed the basis for ISO 12027, commonly called the "SWEBOK Standard" that describes what we now call the classic form of software engineering.

In the process of adopting these ideas, they found that they had to accept the logical positivism that undergirded Shewharts' work. Shewhart was a friend of C. I. Lewis and borrowed ideas from Lewis' pragmatic writings and his critique's of rational positivism. These ideas forced the engineers to recognize that they had to deal with the problem of logical implication, the case in which a true consequent can be paired with a false antecedent and still produce a true statement. While the problems of logical implication could be found in other forms of engineering, it had a power impact on a field what was thought to be based on deterministic automata and logical proof.

---

*Speaker
†Corresponding author: