Chapter 79
Software Engineering Research:
The Need to Strengthen and Broaden the Classical Scientific Method

Gonzalo Génova
Universidad Carlos III de Madrid, Spain

Juan Llorens
Universidad Carlos III de Madrid, Spain

Jorge Morato
Universidad Carlos III de Madrid, Spain

ABSTRACT

The classical scientific method has been settled through the last centuries as a cyclic, iterative process of observation, hypothesis formulation, and confirmation/refutation of hypothesis through experimentation. This "experimental scientific method" was mainly developed in the context of natural sciences dealing with the physical world, such as Mechanics, Thermodynamics, Electromagnetism, Chemistry, and so on. But when trying to apply this classical view of the scientific method to the various branches of Computer Science and Computer Engineering, among which Software Engineering, there are two kinds of obstacles. First, Computer Science is rooted both in formal sciences such as Mathematics and experimental sciences such as Physics, and therefore, an excessive emphasis on the experimental side is not appropriate to give a full account of this kind of scientific activity. Second, the production of software systems has to deal not only with the behavior of complex physical systems such as computers, but also with the behavior of complex human systems (developers interacting with stakeholders, for instance, or users interacting with machines) where educational, cultural, sociological, and economical factors are essential. Therefore, empirical methods in their narrow sense, even though valuable in some respects, are rather limited to understand a reality that exceeds the mere physical world. Moreover, neither formal nor empirical methods can provide a full account of scientific activity, which relies on something that is beyond any established method. Qualitative (i.e. meta-methodical) reasoning plays the directive role in scientific activity. In this chapter, the authors claim that acknowledging a plurality of research methods in software engineering will benefit the advancement of this branch of science.

DOI: 10.4018/978-1-4666-4301-7.ch079

INTRODUCTION

In the final report of the ACM Task Force on the Core of Computer Science, Denning et al. presented an intellectual framework for the discipline of computing intended to serve as a basis for computing curricula (Denning et al., 1989). They characterized computing as a discipline that sits at the crossroads among mathematics, empirical science and engineering. Therefore, there is a *plurality of research methods* that can be applied for the advance of computing in all its branches, one of them being software engineering. The study of mathematics and empirical sciences is recommended in software engineering curricula to promote abstract thinking and the appreciation for the scientific method (IEEE, 2004).

Henri Poincaré wrote at the beginning of the 20th century about the experimental scientific method: "The man of science must work with method. Science is built up of facts, as a house is built of stones; but an accumulation of facts is no more a science than a heap of stones is a house" (Poincaré, 1952, p. 141). A good scientific work is not complete with a systematic recollection of data: it requires a *rational explanation* of their relationships. In other words, the essential ingredients of the scientific method are the description of phenomena (what happens, how it happens) and the explanation of their relationships (why it happens): "What and How describe, only Why explains" (Whetten, 1989).

Michael Polanyi, another scientist-philosopher, claimed that the scientific method is not a recipe that can yield truths mechanically (Polanyi, 1958): explaining the relationships between observed phenomena requires intelligence, imagination, and creativity. Besides, a naïve but widespread empiricist account of science forgets that *observed facts are not independent from theory* (Chalmers, 1999): on the contrary, establishing the validity of observations requires human interpretation and a good deal of previous knowledge. The application of the scientific method in software engineering research has to take into account these warnings. Our main purpose is to show that the advance of science cannot be closed within the application of a "method", however rigorous it may be. There is no universal method in science. In particular, empirical methods are rather limited to understand a reality that exceeds the mere physical world. Each branch of science needs its own method, and selecting the most adequate method is a task that lies beyond any formal or empirical method: it is a meta-methodical task.

In this Chapter we present a brief historical background of the development of the scientific method through the centuries. Then we discuss some philosophical issues that have arisen from the consideration of how scientists work, showing that the scientific method is not fully justified, and perhaps will never be. These issues are relevant when the scientific method is applied to software engineering, as we explain in the last part of the Chapter, where we emphasize the guiding role of qualitative, speculative or conceptual reasoning in research activities.

HISTORICAL BACKGROUND

The scientific method is rooted in the Greek culture, particularly in the laws of logic first defined by Aristotle (384-322 BC). However, Greek science was almost entirely lost for Latin West during the Early Middle Ages, after the fall of the Western Roman Empire. Science was meanwhile safeguarded and cultivated in the Arab culture, with influences from China and India. The main contribution of Indian culture that arrived to Europe through Al-Khwārizmī (c.780-c.850) was the place-value numeral system, which was fundamental for the development of algebraic and arithmetic operations (Mason, 1962). Another significant milestone in the recovery and development of scientific method is the *Book of Optics* by Muslim scientist Ibn al-Haytham (Alhazen, 965-1039), with his pioneering emphasis