Chapter 1
A Comparative Analysis of Software Engineering with Mature Engineering Disciplines Using a Problem-Solving Perspective

Bedir Tekinerdogan
Bilkent University, Turkey

Mehmet Aksit
University of Twente, The Netherlands

ABSTRACT

Software engineering is compared with traditional engineering disciplines using a domain specific problem-solving model called Problem-Solving for Engineering Model (PSEM). The comparative analysis is performed both from a historical and contemporary view. The historical view provides lessons on the evolution of problem-solving and the maturity of an engineering discipline. The contemporary view provides the current state of engineering disciplines and shows to what extent software development can actually be categorized as an engineering discipline. The results from the comparative analysis show that like mature engineering, software engineering also seems to follow the same path of evolution of problem-solving concepts, but despite promising advances it has not reached yet the level of mature engineering yet. The comparative analysis offers the necessary guidelines for improving software engineering to become a professional mature engineering discipline.

INTRODUCTION

Since the early history of software development, there is an ongoing debate what the nature of software engineering is. It is assumed that finding the right answer to this question will help to cope with the *software crisis*, that is, software delivered too late, with low quality and over budget (Pressman, 2008; Sommerville, 2007). The underlying idea behind this quest is that a particular view

DOI: 10.4018/978-1-60960-215-4.ch001

on software development directly has an impact on the software process and artifacts. Several researchers fairly stated that in addition to the question what software development currently is, we should also investigate what professional software development should be. The latter question acknowledges that current practices can be unprofessional and awkward and might require more effort and time to maturate. Although both the questions on what software development is and what professional software development should be are crucial, it seems that there are still no definite answers yet and the debate is continuing from time to time after regular periods of silence. Some researchers might consider this just as an academic exercise. Yet, continuing the quest for a valid view of software development and a common agreement on this is important for a profound understanding, of the problems that we are facing with, and the steps that we need to take to enhance software development.

The significant problems we may face, though, seem not to be easily solved at the level as they are analyzed in current debates. To be able to provide both an appropriate answer to what software engineering is, and what it should be, we must shift to an even higher abstraction level than the usual traditional debates. This view should be generally recognized, easy to understand and to validate and as such provide an objective basis to identify the right conclusions. We think that adopting a problem solving perspective provides us an objective basis for our quest to have a profound understanding of software development. Problem-solving seems to be ubiquitous that it can be applied to almost any and if not, according to Karl Popper (2001), to all human activities, software development included. But what is problem-solving actually? What is the state of software development from a problem-solving perspective? What needs to be done to enhance it to a mature problem solving discipline? In order to reason about these questions and the degree of problem-solving in software development we have first to understand problem-solving better. Problem solving has been extensively studied in cognitive sciences such as (Newell et al., 1976; Smith et al., 1993; Rubinstein et al., 1980) and different models have been developed that mainly address the cognitive human problem solving activity. In this paper we provide the Problem Solving for Engineering Model (PSEM), which is a domain-specific problem solving model for engineering. This PSEM will be validated against the mature engineering disciplines such as civil engineering, electrical engineering and mechanical engineering. From literature (Ertas et al., 1996; Ghezzi et al., 1991; Wilcox et al., 1990; Shaw et al., 1990) it follows that engineering essentially aims to provide an engineering solution for a given problem, and as such, can be considered as a problem solving process. We could further state that mature engineering disciplines are generally successful in producing quality products and adopt likewise a mature problem-solving approach. Analyzing how mature engineering disciplines solve their problems might provide useful lessons for acquiring a better view on what software development is, that has not yet achieved a maturity level. Hence, we have carried out an in-depth comparative analysis of mature engineering with software engineering using the PSEM. In principle, every discipline can be said to have been immature in the beginning, and evolved later in time. Mature engineering disciplines have a relatively longer history than software engineering so that the various problem solving concepts have evolved and matured over a much longer time. Studying the history of these mature disciplines will justify the problem-solving model and allow deriving the concepts of value for current software engineering practices. Hence, our comparative study considers both the current state and the history of software development and mature engineering disciplines. Altogether, we think that this study is beneficial in at least from the following two perspectives. First, an analysis of software engineering from a problem-solving perspective will provide an