Towards Knowledge Evolution in Software Engineering: An Epistemological Approach

Yves Wautelet, Université catholique de Louvain, Belgium
Christophe Schinckus, Facultés Universitaires St-Louis, Belgium
Manuel Kolp, Université catholique de Louvain, Belgium

ABSTRACT

This article presents an epistemological reading of knowledge evolution in software engineering (SE) both within a software project and into SE theoretical frameworks principally modeling languages and software development life cycles (SDLC). The article envisages SE as an artificial science and notably points to the use of iterative development as a more adequate framework for the enterprise applications. Iterative development has become popular in SE since it allows a more efficient knowledge acquisition process especially in user intensive applications by continuous organizational modeling and requirements acquisition, early implementation and testing, modularity,... SE is by nature a human activity: analysts, designers, developers and other project managers confront their visions of the software system they are building with users’ requirements. The study of software projects’ actors and stakeholders using Simon’s bounded rationality points to the use of an iterative development life cycle. The later, indeed, allows to better apprehend their rationality. Popper’s knowledge growth principle could at first seem suited for the analysis of the knowledge evolution in the SE field. However, this epistemology is better adapted to purely hard sciences as physics than to SE which also takes roots in human activities and by the way in social sciences. Consequently, we will nuance the vision using Lakatosian epistemology notably using his falsification principle criticism on SE as an evolving science. Finally the authors will point to adaptive rationality for a lecture of SE theorists and researchers’ rationality.

Keywords: Adaptive Rationality, Bounded Rationality, Epistemology, Knowledge Growth Theory, Requirements Engineering, Software Development Life Cycles

1 INTRODUCTION

Software engineering (SE) is by nature a human discipline. In huge software projects hundreds of analysts, designers, developers, project managers, potential users, etc. are involved in the process of creating a software application aimed to fulfill services easing the everyday work and life of thousands of individuals. To deal with such...
a process the traditional waterfall model in which requirements are collected once for all at the beginning of the project life cycle is nowadays too limited. Researches have led to the definition of advanced modeling languages as well as to more elaborated life cycles using the concept of iterative development. Their success come from the fact that requirements elicitation is no more a standalone task at the beginning of the project but rather a continuous process using feedback loops allowing to better apprehend needs on the basis of prototypes.

In this article we propose a modern epistemological reading of SE as an evolving science as well as software project knowledge evolution. As a first basis, project actors’ reasoning context is studied. Indeed, actors involved in a software project have a limited vision inherent to their bounded rationality. At first sight Popperian theories could be used to describe the evolution of software engineering theoretical frameworks. Due to the nature of SE belonging also to social sciences we will show that this vision is abusive. The Lakatosian falsification principle will be used to demonstrate the complexity of getting a crucial experiment in SE. Adaptive rationality will be used to characterize SE theorists and researchers’ expertise field vision.

This article is organized as follows. Section 2 presents the context of the research, the followed approach and justifies the utilization of the epistemologies used in the article and their combination. It also points to the article contributions. Section 3 proposes an epistemological reading of software processes and on the evolution of knowledge in SE as a discipline as well as within a particular project. To this ends we used Herbert Simon’s bounded rationality, the Popperian knowledge growth theory nuanced by the Lakatosian falsification principle and finally adaptive rationality. The implications are then overviewed in section 4 while conclusions are summarized in Section 5.

2 Problem Statement

This section presents the research context i.e. software engineering and more particularly software development life cycles; the research approach, i.e. the framework developed for applying different epistemologies at different level of a defined software development system. It then turns to the epistemological trends where the application of the particular epistemologies are justified and finally point to the main contributions of the article.

2.1 Research Context

Software Engineering (SE) is an engineering discipline concerned with all the aspects of software production. Following Arlow and Neustadt (2002), a SE methodology 1 is made of a modeling language and a software development process.

The modeling language is the syntax made of concepts associated with visual icons used to build models (see for example the Unified Modeling Language (OMG, 2007; Rumbaugh et al. 1999; Booch et al. 1999)). The modelling language can be:

- Graphical when it uses diagrams with a couple of formalized symbols—representing concepts—linked together through relationships with eventually other graphical artefacts representing constraints;
Discovering Communities of Practice Through Social Network Analysis
www.igi-global.com/chapter/discovering-communities-practice-through-social/25241?camid=4v1a