Chapter I
Organizational Analysis of Small Software Organizations: Framework and Case Study

Jesús Zavala-Ruiz
Metropolitan Autonomous University – Iztapalapa, Mexico

ABSTRACT

The intention of this chapter is twofold. On the one hand, I illustrate the complexity of the small software organization, because it is not a reduced version of a large company. Rather, it has very important advantages and challenges. Then, I use organization studies as a multi-disciplinary and multi-paradigmatic link between disciplines, able to reconcile those distinct visions. On the other hand, I open the discussion on the state of crisis affecting software engineering as a discipline. For that, I try to sensitize the reader to the facts surrounding this crisis, but also to the most promising alternative, which is the redefinition of software engineering as a discipline. One of the possible options for that paradigmatic change requires a multi-disciplinary orientation because their positivist roots and the adoption of a constructivist ontology and epistemology facilitating the inclusion of visions non-qualified for a systematic, disciplined and quantitative approach. My position is that only by opening up this discussion is it possible to begin transforming and consolidating software engineering as a strengthened and more terrain-attached discipline because of its powerful theoretical and practical explanatory capacity.

INTRODUCTION

In modern society, service-related activities are growing in importance. Software is playing a preponderant role. More and more, organizations depend on data-processing processes and on the abilities of highly skilled and scarce personnel. They have been named knowledge workers and this kind of organization is named knowledge intensive firms (Alvesson, 2004, p.5ss) or post-modern organizations. Software organizations are a paradigmatic case of intensive-knowledge
Organizational Analysis of Small Organizations

organizations for many reasons, but principally because its main work is symbolic or non-material in nature and because it is non-standard work for its labour conditions.

Paradoxically, most software organizations are not characterized as applying to themselves what they promote for others. A closer, detailed look inside them frequently reveals a totally different situation: while in the business, most productive, administrative and management processes are candidates for automation, in software organizations, these processes still continue to develop in a craft-like and chaotic manner. The same thing happens in the systems department of any organization. Software production and the most of the work relating to information technology have succumbed to the culture of urgency and competition typical of management. Business cycles decrease and everything is in constant change, with the rising pressure of changing information systems.

Management in software organizations, in a process of imitation, has attempted to adopt the same management recipes applied in business. In this way, it has succumbed to management fads. One of these clearly is total quality management (TQM), adopting the name of the normative standards of that moment. Many TQM initiatives have failed miserably in practice in small businesses as well as in large ones. However, in implementing these organizational models, organizational limits exist for their full application, most importantly in small and medium enterprises (SMEs). Resource constraints in SMEs at all levels (monetary, human, time, opportunity, etc.) make it impossible to adhere completely to such prescriptions. Additionally, those organizational characteristics, such as size, diversity, structure and formalization, impose specific restrictions since causes can be manipulated, but not effects. Possibly, the consideration of formal-informal structure duality is the most important, because it provokes both structures to distance themselves from each other, making a close attachment of the informal one to the formal one practically impossible and expensive. But this structural duality is present in all aspects of the organization and software projects become chaotic because of that, among others. To analyze this organizational complexity, which is magnified in small organizations, a framework has been developed based on organization studies and with a social constructionism approach.

This chapter is organized into three parts. In the first part, I provide an overview of the organizational paradigms that allow estimating the complexity of the organization conceived as a social construct beyond its classical mechanics and simple vision. This way, the organization appears as complex, multi-dimensional and multi-paradigmatic. Additionally, I provide an overview the three disciplines that support software production: management, project management and software engineering. I end with a recounting of facts pointing to software engineering as a discipline in crisis, and the causes of that crisis identified.

In the second part, I propose that if software engineering is in crisis, a paradigmatic change as a discipline is required and I propose three alternatives for that change. In addition, based on Kenneth Gergen’s arguments, I assume that all paradigmatic change can only be possible by means of abandoning the current paradigm, adopting a critical attitude with a new discourse and arguments, and then promoting a new paradigm. This part ends with positing the conceptual tools that constitute a framework for doing organizational analysis in software organizations. It is exemplified by a study case of a small outsourcing software company in Mexico City. In addition, I reflect briefly on the scope and utility of this framework. The application of this framework demonstrates that the use of a distinct paradigm other than software engineering for studying software organizations is possible. Finally, in the third part, I explore some of the main trends I consider promising for research in the short term.