Chapter 6.3
Differentiated Process Support for Large Software Projects

Alf Inge Wang
Norwegian University of Science and Technology, Norway

Carl-Fredrik Sørensen
Norwegian University of Science and Technology, Norway

ABSTRACT

This chapter presents a framework for differentiated process support in large software projects. Process support can be differentiated in different levels based on the size of the development organization and the need for coordination across different levels of the organization. We have defined four main perspectives: individual, group, team, and project level, where the framework consider essential issues when planning and executing the software development processes in organizations with different levels of management. Further, a guideline is provided that suggests what is required of process support in the various organizational levels.

INTRODUCTION

Development of large and complex software systems involves large organisations. In such working environments, it is essential to plan and coordinate the process, feed the involved developers with necessary documents, tools and files, track the process and effort, and learn from and improve the process.

Software process modeling is aimed at understanding, guiding, coordinating, automating, and improving the software process to thus improve the software quality and reduce the effort of developing software products (Wang, 2001). Many ***process models*** and ***process-centred support environments*** (PSEs) have been created with the assumption that the same process support should be provided at every level in an organization (Conradi, Fuggetta, & Jaccheri, 1998; Derniame, Baba, & Wastell, 1998; Finkelstein, 2000; Fuggetta, 2000; Nitto & Fuggetta, 1998).

If we consider development of large software systems, the organisations in such projects usually involve several levels of management. Depending on the level of an organisation a person is working in, the perspective and goal of the work will vary.

For a programmer, the main concern would be to have access to all necessary files, documents, and tools to carry out efficient programming. Personnel working at higher levels in the organisation would typically have other concerns like coordinating people, scheduling of the process, quality assurance, planning of activities and so forth. Thus, it is essential that the process support in such organisations reflects the levels being supported. It is also important that the way the processes are modeled is tailored for the organisational level and the characteristics of this level.

This chapter presents a differentiated process support framework that describes the elements required to model the software process, the required external resources (like tools and documents), and the required process support provided by a process-centred environment. Our framework describes the required process support from four perspectives: At the individual level, at the group level, at the team level, and at the project level. Thus, the objectives of this chapter is to give insights into essential issues to be considered when planning and executing a software development process for large software projects consisting of several levels of management. The chapter also provides a guideline for what is required of process support for the various levels of an organisation. This guideline can be used as input when evaluating tools to be used to support the development and management processes of large software projects.

BACKGROUND

This section gives an introduction to the background and the important terms used in our framework, and describes related work.

Software Process and Software Process Modeling

At a NATO Conference in Garmisch-Partenkirchen in 1968, the term ***software engineering*** was first introduced (Naur & Randell, 1969). The conference discussed what was called the "software crisis" introduced by third generation computer hardware. The work within the software engineering domain has been concerned with methods, techniques, tools, programming languages, and more to face the problems in software projects like delayed products, cost overruns, and bad reliability and performance in software products. Despite the many efforts to try to solve the software crisis, we are still struggling with the same problems as they did in the sixties. Brooks (1986) argues that there is no "silver bullet" that can solve all problems in software engineering. The only way to limit the negative effects identified as the software crisis is to use best practices in the many areas of software engineering. There are no best practices that solve all problems, but in combination many issues can be eliminated. One best practice is to improve the software process itself involving all the activities necessary in developing a software product. This chapter is intended as a guideline for improving the software process at different levels in a software development organisation.

Before we take a closer look at the software process support for different levels of an organisation, it is necessary to agree on the central terminology used in this chapter. As mentioned before, the term *software engineering* covers most aspects involved when developing large software systems. According to Sommerville (1995):

Software engineering is concerned with software systems which are built by teams rather than individual programmers, uses engineering principles in the development of these systems, and is made up of both technical and non-technical aspects.

As Sommerville states, software development involves more than the technical aspects like dealing with the source code of the system. Important nontechnical aspects of developing software involve scheduling, budgeting, resource