

Differentiated Process Support for Large Software Projects

Alf Inge Wang, Carl-Fredrik Sørensen
Dept. of Computer and Information Science
Norwegian University of Science and Technology
N-7491 Trondheim, Norway
alfw@idi.ntnu.no, carlfrs@idi.ntnu.no

August 9, 2006

Abstract

Large software projects involve many actors with various needs related to the development of the software system. Further, large software projects require a software development organisation with management at several management levels. Although the level of hierarchy introduced in the organisation can vary from flat organisations to several levels of hierarchical management, there must be some organisation levels to enable practical project management. Traditionally, process support for software projects have been provided through a standard process interface independent of the level in the organisation. We believe that the process support provided for large software projects must be differentiated to serve each level of the organisation in a proper way. In this chapter, we present a framework to support processes at four distinct levels in a software development organisation: Individual processes, group processes, team processes and project processes. Our framework identifies the process model elements, required external resources and the required process support for these four levels. In addition, we give examples of process models for the four levels.

Keywords: Software process support, software process modelling, process-centred support environment, large software projects.

1 Introduction

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

the process and effort and learn from the process. Software process modelling is aiming at understanding, guiding, coordinating, automating and improve software process in order to improve the software products been developed. Many process models and process-centred support environments (PSEs) have been created usually with the same assumption that the same process support should be provided for every level in an organization [1, 2, 3, 5]. If we consider development of large software systems, the organisations involved in such projects usually involves several levels of management. Depending on the level of the organisation one work, 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 of the organisation would typically have other concerns like coordinating people, scheduling of process, quality assurance, planning of activities etc. Thus, it is essential that the process support for such an organisation reflects the levels of the organisation been supported. It is also important that the way the processes is being modelled are tailored for the level of the organisation and the characteristics of this level. We therefore propose a differentiated process support framework that describes the elements required to model the process, the required external resources (like tools and documents) and the required process support provided by a process-centred environment.

2 Four levels of processes and process support

Here is a short introduction of the four level of processes covered in our framework. The final chapter will go more into details on how each level is modelled, the required external resources (tools and documents), required process support, execution environment, process roles and the interfaces between the four levels.

Individual process: The lowest process level in the organisation is the *individual process level* focusing on the individual tasks of a software developer. A role that is related to this level is a programmer with typical tasks like reading documents, writing code, writing document, debugging code, compiling code, building code, etc. Generally, a process for an individual actor of a software process can typically consist of a set of activities related to the roles that the actor plays. It is very important that the process support is tailorable and configurable, making it possible for the actor to fit the work to personal preferences. The required process support is very dependent on the experience level of the individual actor. An inexperienced person would typically need process guidance that will help this person to know what to do next, what procedures to follow, what tools to use, where to find documentation etc. However, for an experienced person that knows all the basics, extensive process guidance would be more a hindrance to do the work effectively. For the latter, it would be more useful to provide automation of repetitive tasks. Independent of the experience level of the actors, the process support should

provide quick and easy access to the required tools and artefacts to perform the activities. The activities of the individual process should be integrated with personal electronic calendars to enable activity states and deadlines to be accessible on personal computers, personal data assistants (PDAs), mobile phones and even digital watches. This means that the execution environment for the process support also must be capable of communicating and utilizing mobile devices and networks.

Group process: The next level we have recognized in development organisations is the *group process level* focusing at the cooperative aspects of the software process. The cooperative process involves coordination of activities from several actors or roles, cooperative activities where two or more persons must participate at once to create a result, and coordination and negotiation of artefacts and resources. Whenever several persons share artefacts, configuration management (CM) is an important part of the support and coordination of the process. The CM environment must be integrated with the PSE to provide negotiation about conflicting resources and smoothness handover of output artefacts from the process. Further, the group process defines the synchronisation points for the individual processes from several roles or actors. In addition, the group process involves cooperative activities like distributed brainstorming, electronic voting, collaborative authoring, and conflict management. Cooperative activities have very different characteristics compared to individual activities. While for the individual activities the main emphasis is on activities themselves, cooperative activities are all about interaction between roles. This means that the process support for cooperative activities must provide an infrastructure to enable the involved roles to interact in an effective way and to enable flexible exchange of artefacts. Note that most people working in a large software project will be involved at both individual and group processes.

Team process: The *team process level* is related to specific roles of a software project like a team manager and is the lowest management level of the framework. Typical tasks at this level can be planning of team activities, tracking progress of team, monitoring resource usage, and low-level quality assurance. The process support required at this level is monitoring of the on-going processes, computation of resource usage, prediction and estimation of progress and resource usage, and creation and editing of process models. It is also essential that the team process models can integrate the group and individual process models and provide a process environment that will exchange artefacts and state changes between the individual and group processes. The main difference between the coordination and cooperation at the group and team levels is the granularity. At the group level, the cooperation between group members is tight and frequent. At the team level, the groups are independent but the coordination is still important to e.g. manage available resources, and exchange project artefacts (document, source code, programs, tools etc.).

Project process: The *project process level* represents the process of a project at a high abstraction level suitable for project management. For a project manager at this level, the required process support is typically to create a project plan (project model) for the whole project, establish and create teams and groups, assign personnel to teams and groups, estimate project resource usage, track progress of the whole project, monitor resource usage, perform company quality assurance procedures, assess and improve development process, etc. At this level it is not unusual to have external sub-contractors to carry out parts of the project that require special expertise or expertise that is not present in the company. From a process support point of view, it is also likely that external sub-contractors use their own process models and support environment. Thus, it is essential to enable an integration of the process models and support environment to enable monitoring of the whole project.

3 Creation of a Multi-level Process Model

There are two main approaches in order to create a multi-level (4) process model for a software project: *top-down* and *bottom-up*. We will here briefly describe the two approaches (a more detailed description in the full chapter).

Top-down approach: In the top-down approach, typically a project manager will first create a high level process model based on experiences from previous projects and actual information for the current project. This process model should at least identify the activities at the team process level, but contain modelling of group and individual processes. It is important that the individual and group process is not constrained in such a way that it cannot be tailored by the process participants carrying out the processes. If group and individual processes are modelled at this level, the process model should be described as a process template rather than an instantiated process model. If the project manager has not modelled any group or individual processes, the team manager must identify and model individual and group processes. Also here, it is essential that the individual activities are not modelled in such a way that they cannot be altered or tailored by the process participants at the individual process level. The team manager should include sufficient content to the individual process level like activity description, necessary documents and tools and a proposed sequence of the activities. In some cases the sequence of activities must be frozen, because of coordination and synchronisation between developers. The top-down approach is suitable when the top management has a good idea of how the software is being developed in the company or the company management wants to enforce a way of developing software.

Bottom-up approach: The bottom-up approach aims to harvest the process models to the levels below. This means that the team process models will be created

based on the underlying group and individual process models, and the project process model will be created based on the underlying team process models. If no processes are described in the company, the bottom-up approach can be used by gathering information about how the software is being developed and creating process models in order to represent the way you are working. It is likely that the process needs to be simplified to be able to model it.

4 Rest of the Chapter

The rest of the chapter would describe more in details the four level framework including figures describing the framework, a more detailed description of the process elements of the framework (process modelling elements, container elements, interfaces, required external resources, process characteristics, required process support, execution environment, process roles). Further, the chapter will give concrete examples of process models that represent each of the four process levels to describe how each level can be modelled. The chapter will also include a related work section.

References

- [1] Reidar Conradi, Alfonso Fuggetta, and Maria Letizia Jaccheri. Six theses on Software Process Research. In Volker Grün, editor, *Software Process Technology, 6th European Workshop (EWSPT'98)*, pages 100–104, Weybridge, UK, September 16-18 1998. Springer-Verlag. LNCS 1487.
- [2] Jean-Claude Derniame, Badara Ali Baba, and David Wastell, editors. *Software Process: Principles, Methodology, and Technology*. Springer Verlag LNCS 1500, Berlin, Germany, 1998.
- [3] Elisabetta Di Nitto and Alfonso Fuggetta, editors. *Process Technology: Special Issue*, volume 5 of *Journal on Automated Software Engineering*. Kluwer Academic Publisher, January 1998.
- [4] Anthony Finkelstein, editor. *The Future of Software Engineering, 22nd International Conference on Software Engineering (ICSE'2000)*, Limerick, Ireland, 2000. ACM Press.
- [5] Alfonso Fuggetta. Software Process: A Roadmap. In A. Finkelstein, editor, [4], pages 27–34, Limerick, Ireland, June 2000. ACM Press.