

Connecting the Rationale for Changes to the Evolution of a Process

Alexis Ocampo and Martín Soto

Fraunhofer Institute for Experimental Software Engineering
Fraunhofer-Platz 1
67663, Kaiserslautern and Germany
`{ocampo,soto}@iese.fraunhofer.de`

Abstract. In dynamic and constantly changing business environments, the need to rapidly modify and extend the software process arises as an important issue. Reasons include redistribution of tasks, technology changes, or required adherence to new standards. Changing processes ad-hoc without considering the underlying rationale of the process design can lead to various risks. Therefore, software organizations need suitable techniques and tools for storing and visualizing the rationale behind process model design decisions in order to optimally introduce future changes into their processes. We have developed a technique that support us in systematically identifying the differences between versions of a process model, and in connecting the rationale that motivated such differences. This results in a comprehensive process evolution repository that can be used, for instance, to support process compliance management, to learn from process evolution, or to identify and understand process variations in different development environments. In this article, we explain the underlying concepts of the technique, describe a supporting tool, and discuss our initial validation in the context of the German V-Modell XT process standard. We close the paper with related work and directions for future research.

1 Introduction

The field of software process modeling has become established within the software engineering community. An explicit process model is a key requirement for high productivity and software quality. The process description content might be collected in several ways, for example by observing real projects, describing intended activities, studying the literature and industry reports, or interviewing people involved in a project [10]. Usually, considerable effort is invested into the definition of such processes for an organization. Once the process is defined and institutionalized, modifying it further becomes unavoidable due to various reasons, such as the introduction of a new software development technology in a development team (e.g., new testing support tools and techniques), a new/updated process engineering technology (e.g., a new process modeling technique), new/updated standards/guidelines for software development or process engineering, new/updated regulatory constraints, or new/updated standards/guidelines for software development or process engineering. Such changes must be reflected accordingly in the corresponding process models. Achiev-

ing a compromise that satisfies such a challenge usually depends on the information available for rapidly judging if a change is consistent and can be easily adopted by practitioners.

Having information about the reasons for process changes (i.e., the *rationale*) at hand can be of great help to process engineers for facing the previously mentioned challenges. Currently, the common situation is that there is a lack of support for systematically evolving process models. Combined with other facts such as budget and time pressure, process engineers often take shortcuts and therefore introduce unsuitable or inconsistent changes or go through a long, painful update process. In many cases, precipitous and arbitrary decisions are taken, and process models are changed without storing or keeping track of the rationale behind such changes.

According to our experience, systematically describing the relationships between an existing process and its previous version(s) is very helpful for efficient software process model evolution [2]. Such relationships should denote differences between versions due to distinguishable modifications. One can identify the purpose of such modifications if one can understand the rationale behind them. Rationale is defined as the justification of decisions [8]. Rationale models represent the reasoning that led to the system or, in our case, to the process in their current form. Historically, much research about rationale has focused on software/product design. By making rationale information explicit, decision elements such as issues, alternatives, criteria, and arguments can improve the quality of software development decisions. Additionally, once new functionality is added to a system, the rationale models enable developers to track those decisions that should be revisited and those alternatives that have already been evaluated.

We are currently working on transferring rationale concepts into the process modeling domain. We do this based on the assumption that the rationale for process changes can be used for understanding the history of such changes, for comprehensive learning, and for supporting the systematic evolution of software processes. We are looking at the possibilities that can be used for documenting changes and connecting them to their corresponding rationale.

This article presents a technique for comparing process models, recognizing a set of standard changes, and connecting them to their respective rationale as follows: In Section 2, we present the conceptual model for capturing rationale; In Section 3, we present the technique for identifying changes. In Section 4, we illustrate the connection of changes to rationale; In Section 5, we briefly discuss our implementation of this technique, as well as our experience in applying it to the German V-Modell XT [33] process standard. In Section 6, we present a short description of related approaches for comparing models and for capturing rationale; and finally, in section 7, we provide a summary and future research questions to be resolved.

2 Process Rationale

The following is a conceptual model that can be considered a second version of our attempt to understand the information needs for capturing the rationale behind process changes (see Fig 1). The results of our first attempt have been documented in [21]. In