Chapter VI

Method Chunks to Federate Development Processes

Isabelle Mirbel, I3S Laboratory, France

Abstract

Method engineering aims at providing effective solutions to build, improve, and support the evolution of development methodologies. Contributions in the field of situational method engineering aim at providing techniques and tools allowing one to build project-specific methodologies. However, little research has focused on how to tailor such situational methodologies when used as organization-wide standard approaches. Moreover, current approaches have been thought of for method engineers, that is to say, expert users, and they are not enough dedicated to nonexpert ones. In this context, we propose an approach that consists of federating the method chunks built from the different project-specific methods in order to allow each project to share its best practices with the other projects without imposing to all of them a new and unique organization-wide method.
Introduction

Several decades of work have been spent to provide effective solutions to build, improve, and support the evolution of development methodologies. Different approaches have been successively proposed to provide suitable support to software-based information system development. Experiments show that the provided models and methodologies have been adapted to each of the different situations in which they have been used. At the end, almost every project has carried out tailoring in order to apply effectively best standard practices. There exist now a lot of variations around a given methodology, each of them appearing suitable for the situation (i.e., the organization or the project) it has been customized for, but they are not so easily translatable in a somewhat different situation, even inside the same domain (i.e., the application domain or the organization).

A development methodology (or process) may be seen as a transformation process (where nonformal specifications are transformed into more formal specification and then code), a decision-making process (where the taken decisions are recorded all along the development process), or a problem-solving process (where solutions are provided to the successive problems encountered during the development process). Especially with regard to these two last viewpoints (decision-making and problem-solving aspects), it would be interesting to benefit from the experiences acquired during the resolution of previous problems. Moreover, the rich and long experience we already have in supporting software-based information system development leads us to try to capitalize and share best practices in this field as it has already been successfully done in the software development domain.

Indeed, there is a need for the capitalization and sharing of knowledge about method engineering as well as a need for customization and tailoring of this knowledge to be better adapted to the organization, the project it is deployed in, and even the user it is targeted for. In this chapter, we start first by discussing the proposals made in the field of method engineering (and especially situational method engineering, aiming at providing solutions to customize development methodologies) and the work done on software reuse. Then we show the shortcomings of the provided approaches. As will be detailed, current approaches have been thought of for expert users and not enough are dedicated to nonexpert ones. Moreover, they are not very suitable when used as organization-wide standards.
A Framework for Managing Complexity in Information Systems
www.igi-global.com/article/framework-managing-complexity-information-systems/181668?camid=4v1a

Incorporating Spatial Queries into Semantic Sensor Streams on the Internet of Things
www.igi-global.com/article/incorporating-spatial-queries-into-semantic-sensor-streams-on-the-internet-of-things/194998?camid=4v1a