A Software Engineering Environment (SEE) is quite useful in order to manage the complexity of SM projects, since it can provide the needed services. Of the different aspects to highlight in these environments, in this chapter we put our main attention on those that are more directly related to the goal of helping in the management of SM complexity: to approach the SMP from a wide perspective of business processes to integrate technological and management aspects; to define a Process-centered Software Engineering Environment (PSEE); and to use a multilevel conceptual architecture based on standards like MOF (Meta-Object Facility). The MANTIS proposal of integral environment for the management of SM projects is also presented, and the main components of this environment are commented: conceptual tools (multilevel architecture, ontologies, software processes models and metamodels); methodological tools (methodology, and interfaces with organizational and managerial processes) and technical tools (horizontal and vertical software tools, repository, and interaction with process enactment software tools).
In the area of Software Maintenance (SM), there are still a number of matters to study and research (Bennett & Rajlich, 2000). One of the most important is the development of tools and environments to support methodologies and to facilitate the reuse of processes (Harrison, Ossher, & Tarr, 2000). A Software Engineering Environment (SEE) is quite useful to manage the complexity of SM projects, since it can provide the needed services. The SEE must be capable of managing data and metadata of the different production processes—in our case, the Software Maintenance Process (SMP)—at different detail and abstraction levels. The SEE should be based, for this purpose, upon a conceptual multilevel architecture, allowing all the information of processes to be shared among all available tools. This last need is satisfied using a repository manager that saves data and metadata of processes using an open and portable format.

In this chapter, a conceptual multilevel architecture is presented, making the integration of all available tools for managing SM projects possible, in precisely an integrated environment.

The fact is that such a SEE is a help to approach the inherent complexity of the SMP from a broader perspective than the merely technological one.

A SEE must satisfy several requirements to reach the aforementioned general goal. The two most meaningful requirements are the following: it must be process-oriented, and it must permit work with different models and metamodels of the software processes involved in the SM projects.

Of the different aspects to highlight in these environments, in this chapter we put our main attention on those that are more directly related to the goal of helping in the management of complexity. The first section presents a proposal to approach the SMP from a wide perspective of business processes, integrating technological and management aspects. The use of a Process-sensitive Software Engineering Environment (PSEE) to reach such goal is justified in the second section, and its architecture is presented.

The importance of using a multilevel conceptual architecture are justified in the third section and how to apply the Meta-Object Facility (MOF) Standard to the SM is commented. The MANTIS proposal of integral environment for the management of SM projects is presented in the following section. Lastly, the main components of MANTIS are commented in the remaining sections: conceptual tools (multilevel architecture, involved processes, ontologies, and metamodels); methodological tools (methodology and interfaces with organizational and managerial processes) and technical tools (horizontal and vertical software tools, repository, and interaction with process enactment software tools).
Related Content

All-Optical Internet: Next-Generation Network Infrastructure for E-Service Applications
[www.igi-global.com/chapter/all-optical-internet/29573?camid=4v1a](www.igi-global.com/chapter/all-optical-internet/29573?camid=4v1a)

On the Identification of Modeler Communities
[www.igi-global.com/article/on-the-identification-of-modeler-communities/112040?camid=4v1a](www.igi-global.com/article/on-the-identification-of-modeler-communities/112040?camid=4v1a)

A Comparative Analysis of Reliability Assessment Methods for Web-based Software
[www.igi-global.com/article/a-comparative-analysis-of-reliability-assessment-methods-for-web-based-software/103280?camid=4v1a](www.igi-global.com/article/a-comparative-analysis-of-reliability-assessment-methods-for-web-based-software/103280?camid=4v1a)
Software Product Lines to Faster Achieving Productivity

www.igi-global.com/chapter/software-product-lines-faster-achieving/37028?camid=4v1a