The Development Process

The Web technologies discussed in the previous chapter enable the development of Web applications ranging from small, ad hoc solutions to complex Web information systems. Before focusing on the actual development of such kinds of applications, i.e., the products, we would like to focus on the process that leads to the creation of a Web application. Understanding software development processes (in general, not only for Web applications), the main development activities that need to be performed, their interconnections, and their temporal order are of fundamental importance for the success of a software product. In this book we follow a top-down approach to the organization of the contents and discuss first the way (Web) application development is organized, so as to discuss then the single activities that we will identify in this chapter.

Note that if we shift our point of view from that of the developer or project manager to that of the software product, what was before a software development process can now be seen as software life cycle, that is, a model that describes the life of an application, from its inception to its dismissal. Software development process and software life cycle are thus synonyms used in the literature depending on which view one prefers to highlight. As this is a book about Web engineering, we will use more the term software development process, though software life cycle may be used as well.

In this chapter, we will first discuss software development and its processes that are generally executed for any software product, in order to introduce the reader to the basic concepts and activities. We will then describe a possible development process more specific to Web applications and discuss its differences with more traditional development processes. We will also introduce some examples of concrete Web development processes, in order to introduce the reader with the peculiarities of the Web. The rest of the book will then be structured according to a typical process model.
3.1 Decomposing the Software Development Process

In today’s software industry it is hard to find products that are planned, implemented, and tested by a single developer, as the complexity of modern (Web) applications typically requires the involvement of several different experts who are able to address specific development requirements more precisely. Depending on the size of the application and the actors involved in the development process, building an application may be an intricate undertaking, exposed to a variety of risks that might compromise the success of the final application. In order to control the software development process, it is thus of fundamental importance to understand its constituent activities, its actors, and their interconnections.

3.1.1 Activities in Software Development

Software development is a creative process leading to an innovative software product or system. Usually, this process is not just one monolithic block of work that takes as input some ideas about the application to be developed and produces as output a perfectly fitting solution; the process can be decomposed into a set of basic activities with well-defined boundaries and meanings. Such activities aim at understanding the problem, planning a solution, carrying out the plan, examining the result for accuracy, and resolving possible errors or inaccuracies. Traditionally, the software development process is organized into the following basic activities:

- **Requirements engineering**: aims at understanding the problem.
- **Design**: aims at planning a solution to the problem.
- **Implementation**: translates the plan into running application code.
- **Testing and evaluation**: aims at identifying coding errors or inconsistencies between the collected requirements and their implementation.
- **Deployment**: brings the solution to the customers.
- **Maintenance**: aims at monitoring a running system and keeping it healthy and running.
- **Evolution**: aims at improving the developed solution over time, providing new input to the development process in the form of new requirements.

More precisely, **requirements engineering** aims at understanding a product’s needed capabilities and attributes. The analysis concentrates on **functional requirements**, referring to the functions that the system must be able to support, as well as on **nonfunctional requirements**, referring mainly to the quality of the offered solution. This implies identifying the general idea behind the system, as well as the stakeholders that require the new solution, the motivations for the production of a new system and the final usage environment. The collected requirements are elaborated with the aim of producing some high-level models of the system that abstract from irrelevant details of the problem domain.