Software architectures patterns and its relation to software development methodologies

Abstract

Nowadays, software engineering refers to the science of designing, implementing and modifying the software so that it has higher quality, greater productivity, maintainability and more speed of construction. Hence, the systematic approach to analyzing, designing, evaluating, implementing, testing, preserving, and reengineering are considered as important applications in software engineering. Software architecture is one of the most important topics in the development of software engineering. In fact, by expanding the area of software systems, the main challenge ahead is the development and improvement of software architecture as the basis of software systems. Software methodologies are not apart from this fact. The importance and necessity of methods, their maintenance, the training of human resources and the promotion of methods will make the role of architecture more vital, especially in the methodologies that have been less attention to them.

In this thesis, we try to understand the interaction between software methodologies and architectural patterns and address the relationship between them. In addition, while taking advantage of an architecture-based design and study of the benefits of architectural patterns as well as styles in software development, we will try to find a way to use their benefits. Same as the methodologies for creating software in different projects, the used software architectures have common and different aspects. Considering an architecture for all projects is not feasible, because each project has unique features in accordance with its conditions. But the common features of different architectures can be extracted and with their detailed description, the architecture of the project based on these defined features can be determined. Using styles reduces the cost of software architecture and increases the accuracy of it since there are some guaranteed quality features for each architectural style. Styles include features such as reuse, documentation, early detection of risks and an easier update for the project. Obviously, there are some negative qualities for the styles which can be used due to the project conditions.

Keywords: Software methodologies, software architecture, architectural styles.