ABSTRACT

In today’s world of rapidly advancing technology, guaranteeing software quality is of paramount importance. The quality of software is intricately connected to the underlying architecture. It is recognized that it is not possible to measure the quality attributes of the final system based on the software architecture design alone, because the software architecture of a system is defined as the “meta-structure”, “which comprises software components, the externally visible properties of those components, and the relationship among them”. This definition focuses only on the internal aspects of the system. In the current work, analysis models such as SAAM (Software Architecture Analysis Method), SAAMCS (Software Architecture Analysis Method Founded on complex Scenario), ESAAMI (Extending SAAM by Integration in the Domain), SAAMER (Software Architecture Analysis Method for Evolution and Reusability), ATAM (Architecture Trade-Off Analysis Method), SBAR (Scenario-Based Architecture Reengineering), ALPSM (Architecture Level Prediction of Software Maintenance), and SAEM (Software Architecture Evaluation Model) were considered.

The Architecture Trade-Off Analysis Method (ATAM) was found the best among them for evaluating the software quality attributes. But, it suffers from inherent drawbacks in terms of cost, quality attributes and business focus. Hence a better strategy called EATAM (Enhanced
Architecture Trade-Off Analysis Method) was proposed by combining the features of ATAM with design patterns. Here, a software architecture model is being presented which will overcome the drawbacks in ATAM, and EATAM by using various innovative patterns like Subtraction pattern, Multiplication pattern, Division pattern, Task Unification pattern and Attribute dependency change pattern. The various innovative patterns were applied to the case studies and their results were discussed.