Nowadays in the industry, there is a high demand of software systems with the increase in complexity and facing the challenges of quality requirements, which emphasizes on architecture and implementation mechanism.

The proposed approach helps the software architects to find an efficient way of dealing with requirements related to quality attributes of the system for implementing the suitable design pattern. The Architecture forms the basis of achieving quality attributes of the system.

Quality attributes are non-functional requirements of products which helps stakeholders to judge the quality of the system. Some of the quality attributes include usability, availability, configurability, interoperability, portability, performance, modifiability, security, reliability etc. Degree of a system to meet the quality attribute requirements is dependent on its design and architecture.

Decision making has an important role in the creation of Software Architecture. Such decisions are required throughout the software development life cycle and to attain the quality attributes of a software product. Decisions which are made during architectural design have direct impact on software quality. Many design patterns are available in software architecture and these design patterns help achieve quality attributes in a software product but decision-making for the solution of these design patterns is still an issue. In this paper, a new perspective of selection of design pattern for a given problem is presented. PRIC (Problem, Requirement Identification, Compatible design pattern) model is proposed which helps decision-making of selection of design pattern to attain the required quality attributes.

PRIC Model: This model explains the proposed process of decision making in Software Architecture. PRIS Models has been shown in Fig. 1. Main phases of this process are as follows:

**A. Problem**

Understanding problem is the most important part of any software Architecture process. Layout of Software Architecture depends on identifying and understanding problem statement.

**B. Requirement Gathering**

After understanding problem, requirements are gathered. Requirement elicitation is one of the main things in Software Architecture and Development. Most of the software Projects fail only because of poor requirement elicitation. Gathering complete requirements using proper requirement elicitation techniques is one of the main success factor in Software Projects.

**C. Identifying Functional & Non- Functional Requirements**

After Requirement Elicitation one of the most important task is to identify and categorize functional and Non-functional requirements, so that decisions are taken accordingly.

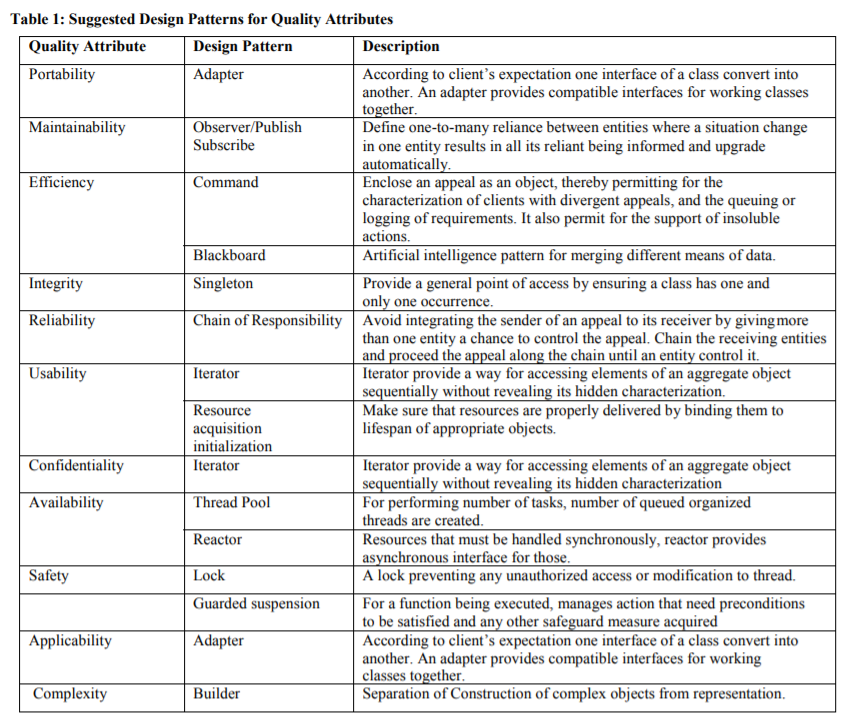
**D. Identifying Quality Attributes.**

Quality attributes are identified after separately identifying functional & Non- Functional requirements. Quality attributed are identified and prioritized on the basis of requirements.

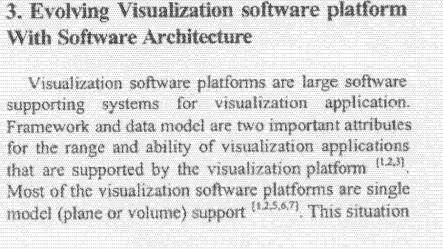
**E. Compatible Design Patterns for Quality Attributes**

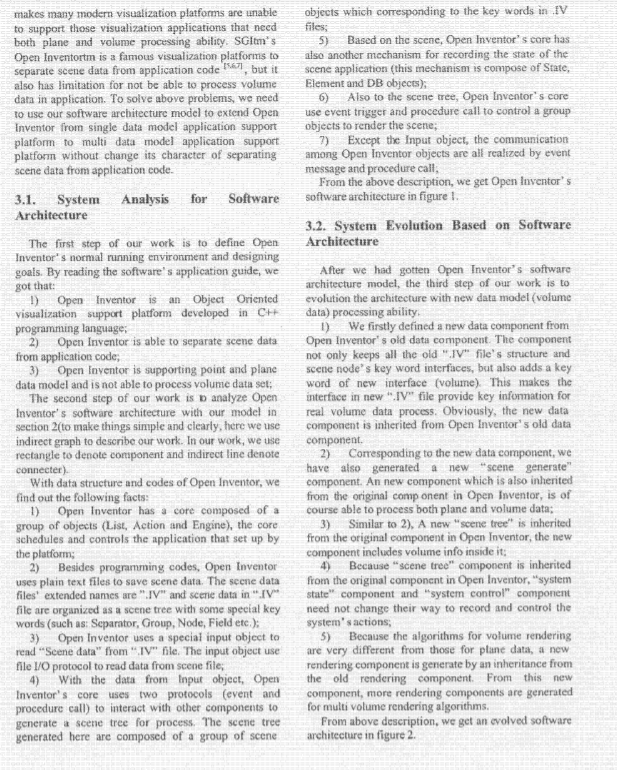
Selecting Compatible Design Pattern according to identified quality attribute is the main decision. Choosing wrong design pattern can cause Software failure, but if chosen rationally it can help out and ease the task of Software Architect.

To explain the further applicability and importance of PRIC Model related to some quality attributes is being discussed along with design pattern suggested for each attribute in Table 1. which will make the process of decision making easy in Software Architecture phase



2)





**Conclusion:**

* Software architecture plays an important role in the construction and maintenance of software.
* If the software architecture is properly developed it will help in the discovery of architectural bad smells, i.e., architectural choices that have detrimental effects on system lifecycle properties.
* Well developed architecture enables more reliable assessment of system quality attributes like performance, security, interoperability, reliability, availability.

Hence, in this paper, a model named PRIC is proposed. This model presents the process of problem identification, requirement gathering, extraction of function and non- functional requirement, and then identifying quality attributes along with the selection of suitable design pattern according to the required quality attributes. Since every software starts from a specific problem domain, it is dependent on the software developers to identify the problem, specify the requirements from the problem domain according to the customer’s satisfaction.