CHAPTER 4

IMPORTANCE OF SOFTWARE ENGINEERING

4.1 INTRODUCTION

The software engineering has been related to many fields and disciplines. A literature survey has been done in the field of software engineering so that the importance of this field can be understood. The focus is on knowledge engineering which encompasses Artificial Intelligence, Computational Intelligence, Knowledge bases, Data mining and machine learning. A number of typical examples of software development issues that can be benefited from these disciplines have been identified. For the sake of clarity and discussion, the literature survey has been broadly classified into four broad categories. They are given below:-

- Planning, Monitoring and Quality Control of Projects
- The Quality and Process improvement of software organization
- Decision Making support
- Automation

The First category that is planning, monitoring and quality control of projects fully depends on the entire past project data. Many ideas have been taken from the past project data to plan and monitor the software development process. But, the planning of the software development process is more difficult if the project is entirely adhoc. Many techniques from other disciplines which help in the above - said processes are Machine Learning, Computational Intelligence and Knowledge Based systems, Stukes et al
(1996). The most important concept in the field of software development is that “Software requirements will change often”. An important phrase says that “Changing is the only constant in the field of software”.

Second focus is that software organizations are inherently learning organizations, which will adapt to this changing field based on experience and project data and project feedback. In order to cope up with this changing environment, software organizations collect numerous amounts of data for the projects. The main question here is that how this information is used for future purposes and also how it is stored in the database.

Third important step in the software development process is that the project managers should make decisions based on the project. So, the decision making is a very important factor and it should be done when the project is in a critical state. Many decision – making tools are available in the market to support the decision making process effectively. But the question that arises here is how far these commercial tools are reliable in making the decisions.

The last category of applications which support the principles of software engineering is automation, Wai – Ming et al (2002). Many automation tools have been developed to support the principles of software engineering so that the task can be finished effectively with less time. But, again the main problem is whether the automated tools support all the projects developed in the organizations. By taking into consideration the above - said categories a hybrid tool has been developed to estimate the effort.

4.2 PLANNING, MONITORING AND QUALITY CONTROL

Like any human – intensive engineering activity, reliable techniques should be developed to plan and monitor the quality aspects of the software developed activities, Philippe Du Bios (1995). More precisely,
project expenditures need to be predicted and significant deviations need to be monitored. This requires the construction of accurate prediction models to take significant deviations and take remedial actions. So, by taking these points into consideration, a hybrid tool has been proposed and its advantages are given below:

- It can easily handle qualitative, categorical data, which are common in software engineering.

- It produces models that are easier to interpret, which is important in our case as we would like to understand what factors affect software development productivity and quality.

Existing Technologies for Planning, Monitoring and Quality Control:

1. Computational Intelligence

   Neural networks are good at building complex problems. They do not require any assumption regarding the functional form of the relationships between predictors and the variable to be predicted. However, their usage may be tedious and the interpretation of the resulting models is difficult. So, in practice, it is very difficult to implement.

2. Fuzzy Set Theory

   Fuzzy set theory has also been used to help software engineering prediction models. The main motivation behind this model is that the data which rely on prediction models are qualitative and subjective in nature. So, this model is restricted to a set of programs whose data should be in the form of categories.
3. **Case Based Reasoning (CBR)**

This Case Based Reasoning strategy has been used in the context of quality and cost prediction for the software to be estimated. The basic principle that lies behind CBR is to define a similarity and it is used to retrieve similar projects so that its cost or quality data can be reused as a basis for prediction. The main drawback of this CBR model is that a similarity function should be defined beforehand. But in software engineering, it is very difficult to define such a similarity function before estimation.

Ideally, software engineering prediction models should combine expert opinion and project data. One important question is then how to integrate expert opinion and project data into common models.

### 4.3 SOFTWARE LEARNING ORGANIZATIONS

Within an organization, experience and knowledge acquired on past software projects can be used to improve practices on future projects. Different issues have to be addressed before making this possible:

- **Technical Issues:**

  Data and documents need to be stored and retrieved in an efficient manner. Knowledge bases need to be designed and maintained and connected to the company intranet for corporate – wide accessibility. As a result Security issues then arise as some of the information may be confidential.

- **Organizational Issues:**

  Such knowledge bases need to be fed by projects. Data, information and documents need to be provided in a consistent form, based on agreed – upon structure and content. The information provided must be precise,
accurate and complete. This required a certain organizational discipline with procedures that are defined and enforced.

- **Cognitive Issues:**

  Users accessing such knowledge bases may be faced with tremendous amounts of information, most of them being irrelevant to the problem at hand. It is therefore important to reduce the cognitive load of the user by allowing him to retrieve, in an efficient and precise manner, relevant information.

  In this context, the design and maintenance of corporate wide knowledge bases become a key issue to address.

4.4 **AUTOMATION**

Many activities in software engineering need to be automated so as to make methods and techniques economically viable. There are many techniques available to address a number of important software engineering issues, Charette et al (1989). Though few techniques and problems of software engineering have been discussed, it is clear that this study only scratches the surface. The potential for cost-effective applications in software engineering is enormous.