ABSTRACT

The basic aim of software fault prediction is to identify error prone tasks as the cost can be minimized with advance knowledge about the errors and this early treatment of error will enhance the software quality. Case-based reasoning is used to predict software quality of the system by examining a software module and predicting whether it is faulty or non faulty. In this thesis an attempt has been made to propose a model with the help of previous data which is used for prediction. Five different similarity measures, namely, Euclidean, Canberra, Exponential, Clark and Manhattan are used for retrieving the matching cases from the knowledgebase. The use of different similarity measures to find the best method significantly increases the estimation accuracy and reliability. Based on the research findings in this thesis it can be concluded that applying similarity measures in case-based reasoning may be a viable technique for software fault prediction. In addition to software fault prediction, this thesis attempts to develop a system to predict rate of improvement of the software quality at a particular point of time with respect to the number of lines of code present in the software. Having calculated the error level (EL) and degree of excellence (DE) at two points in time, we can move forward towards the estimation of the rate of improvement of the software quality with respect to time. This parameter can be used to judge the amount of effort put into while developing software and can add a new dimension to the understanding of software quality in software engineering domain. The effort of this thesis is also directed towards introducing a new mathematical model to understand the state of quality of software by calculating parameters such as the time gap and quality gap with relation to some predefined standard software quality or in relation to some chalked out software quality plan. This thesis also indicates methods to calculate the difference in the quality of the software being developed and the model software which has been decided upon as the criteria for comparison. These methods will provide a better understanding of quality as compared to other standards. The thesis also presents some new ideas about estimation and evaluation of the quality of software. At the outset, it deals with the possibilities of using a standard conversion method so that lines of code
from any language may be compared and be used as a uniform metric. It also attempts to explain in depth the method of evaluating and understanding quality with respect to development time as well as LOC. The present work is also credited through the introduction of some new terms like efficiency and variation to understand the change in software quality. The main focus is to evaluate and estimate software quality at a particular stage of software development. This is not average quality understanding, but quality estimation at a particular instance. One of the salient aspects of the method suggested is that the developer can evaluate the work at any stage using the methods given to review the present status and make future plans to meet the required target. Different types of variations have also been outlined in the thesis. This thesis covers the different kinds of graphical shapes that may arise out of possible cases and gives their respective interpretations.

Keywords: Case-Based Reasoning, Software Quality Prediction, Analogy, Similarity Function, Machine Learning, Knowledgebase Building, Expert System, Accelerated and Retarded Variation, Time Gap, Quality Gap, LOC, Degree of Excellence, Error Level, Testing.