Abstract

Due to recent growth in size and complexity of software system, software development with acceptable level of reliability within given time and budget constraints has become a challenging objective. This objective could be achieved to some extent through early prediction of number of faults present in the software. A major cost of software is incurred during testing phase of the software development. Besides, optimal allocation of test resource allocation and scheduling is desirable to achieve certain level of reliability of software in a cost effective manner. Therefore, this study aims to develop models to predict number of faults during early stage of software development, existing number of residual faults in the software, and classification & ranking of software modules based on their degree of fault-proneness. A number of models are existing in the literature for predicting number of faults present in the software. However, process maturity level, which is a strong process quality indicator, has not been widely addressed along with other significant process quality indicators. Therefore, effect of process maturity along with other quality metrics on number of faults is studied to provide a model for early prediction of range of number of faults during requirement, design, and, coding phase of software development, respectively. This work also presents another approach to predict residual faults in software through a multi-stage model. The proposed model uses generic fuzzy profiles and rules to extract fault density indicator for each phase of software development. Large and complex software containing a large number of modules need to be tested optimally by prioritizing test resources based on fault-proneness of modules. This work proposes a model to classify the modules into two categories a) fault-prone and b) not Fault-prone. Besides, it predicts degree of fault proneness on a scale of zero to one, and ranks the module accordingly. Iterative Dichotomiser 3 algorithm is being used for deriving relationship among software metrics to classify software modules as fault-prone or not fault-prone.

In brief, this proposed work in tends to develop models, which reduces the effort required in development of fuzzy profiles and rules, which is a tedious task when there are numerous software metrics are being applied. Proposed model is easy to use and implement for achieving software reliability cost effectively.

Keywords: Software reliability, Metrics, Faults, Fuzzy inference system.