Abstract

Due to dependability on the computer system for most of the daily activities, one has to trust on the application of the software system. Real time applications need reliable software and reliable software can be produced by controlling the induction of faults at each stage of development life cycle. More early is the estimation to control the faults better will be the results. As object-oriented approach is an effective and efficient way of achieving the basic goals of software industry i.e. to produce reliable products within the limits of cost and time. Some metrics have already been developed for various phases of object-oriented software development and also especially for design phase, but only a few of them use the data available at design phase and maximum of them use the source code information. Thus we still need more metrics of early phases of development to optimize the strength of object approach. The design phase is a focus area as it is the earliest phase to depict the structural details of product to be implemented in reality during its development. In the present study attempt has been made to propose the object-oriented software metrics which would be helpful in early prediction of faults induced in a class using design time details to acquire the reliability using fault forecasting techniques.

To achieve the goal of the study the entire research work is divided into seven chapters; chapter one entitled “Introduction” functions as the blue print of the entire work. It begins with an introductory note followed by the logical successive link of the work. The importance of the object-oriented design is highlighted along with the background information of software engineering. The basic concept behind the object-oriented software design approach is to save cost, manpower and even severe consequences of software disaster. Further the basic research questions are mentioned in the form of research hypotheses. A comprehensive list of objectives of the work is also propagated which can provide sense of the work in brief. It also includes the originality and significance of the work. Chapter second presents the review of literature to acquire the knowledge of past and to show the current trends in the field. The review covers six broad subject domains involved with the present research problem; third chapter provides the overview of whole research process and methods/techniques used at various stages. In chapter fourth the most popular existing metrics Weighted Methods per Class (WMC), Response for a Class (RFC), Coupling between Objects (CBO), Depth of Inheritance
(DIT), Number of Children (NOC), Lack of Cohesion of Methods (LCOM) and Source Lines of Code (SLOC) in software industries and research organizations used for measurement in object-oriented environment are empirically analysed to predict the faults at class-level in early stages. The metrics and faults data from 145 classes of KC1 project is used for analysis. This is a public NASA data set available at http://mdp.ivr.nasa.gov; in chapter fifth four design metrics namely CMCM (Class Member Complexity measure), CICM (Class Inheritance Complexity Measure), CALM (Class Aggregation Level Measure) and CCOM (Class Cohesion Measure) are proposed to measure the class-complexity induced by the various design concept of object-technology i.e. data-hiding, inheritance, aggregation and cohesion. Cognitive theory is taken as theoretical basis for the development of design metrics or in other words cognitive theory is used for justifying the development of design metrics at proposition time as they show association with cognitive complexity. Chapter six focuses on software reliability, induced faults and reliability, fault-forecasting technique to acquire reliability, analysis of design metrics using cognitive theory to observe association with induced faults and empirical analysis is done to authenticate the theoretical basis of metrics development depicted by cognitive analysis results and to find the level and mode of association along with to develop best faults prediction model. Next to these, faults prediction model is further validated through spearman’s correlation coefficient and its utility is described to acquire some level of product reliability in early phases of development (before the coding starts) through redesign by identifying classes having more faults. Last chapter reveals the concluding remarks of the research. Here the scholar describes her own research results and how it engages with the research questions. Further the objectives will be highlighted along with the research hypotheses underlined for this work and also how these hypotheses are proved or disproved. Impact of new metrics is analyzed. At last the research trend is described and how further research will be a beneficial in this specified area are also portrayed.