Abstract

The criteria or procedure of software development is going to be changing continuously by changing time. Some of the new functions may be added, modified or deleted according to the requirement of the users. Through the continues changes, the concept of software evolution is originated. These regular changes in any software during development are a very complex process. As the presence of same code more than one time is if increases then the quality of that particular software are automatically degraded. As founded by the authors in their previous study the maximum use of copy paste programming during software development may also effect on some other factors as an example software maintainability, software reusability, software performance, maintenance cost and overall software performance. The presence of code clones makes the software maintenance extremely difficult. So, to detect software clones during the development of any software or after the development of that particular software is a mandatory or priority for any software developer. Code clones identification thus becomes extremely necessary for improving the maintenance, reusability, performance and in other words, you may say for improving the overall quality of any software. Several studies show that
about 5% to 20% of software systems can contain duplicated code even without doing little or minor modifications. As I studied in my journey of research there are various types of techniques are available in the software industry which are used for software clone detection viz. Token based Approach, Line based Approach, Lexical based Approach, Program Dependency Method, Abstract Syntax Tree based Method, Text Based Method and Metric Based Method. Every technique has following different criteria for detecting different type of clone viz. Type 0, Type 1, Type 2, Type 3 and Type 4. Every type of software clone lies in its own distinct software class so that can be described with different definitions as an example Type 0 performs exact cloning, Type 1 clone is used to detect identical copies except comments, Type 2 clone is used to detect some of the user defined names as like literal names and function names, Type 3 clone is used to detect added, deleted or interchanged lines and Type 4 Clone is used to detect unintentional or un-knowingly presence of similar code or in other words some software developers say auto-generated code which is too difficult to detect. This research paper has focused on to detect and eliminate the software code clones that are present in any software more than one time. I propose a new methodology which is termed as “A Hybrid Technique for software code clone detection by using Token Based and Line Based Approach” (HTSCCDUToLiA) whose main purpose is to detect different types of clones viz. Type 0, Type 1 and Type 2 clones etc. This newly designed methodology working is based on two different types of software cloning approaches viz. Line based Approach and Token Based Approach. This new designed hybrid method will produce more efficient results than already existing techniques. The major objective of this paper is to remove redundant code or free space which is covered by the comment lines especially. The main significance to propose this hybrid technique is to automatic detection of different software code clones within minimum duration of time. Different parameters are considered for software code clone detection in different tables as an example size of code, type of clone, efficiency and portability etc. In addition, at the last the percentage of code clone detection is also calculated by utilizing a different comparison parameter. In addition, the major benefit to design this new hybrid technique is to save software developer time, computer memory space as well as developer effort. By utilizing this new designed methodology the amount of code clones under a specific project or specific application can be easily reduced or removed up to some extent that will ultimately increases the overall performance of the software. In this way, this new designed methodology, in future will helpful for producing more consistent or more efficient results. And hence, the different software parameters viz. software reusability, software maintenance, software performance and overall software quality can be easily improved and easily managed.

References


**Index Terms**

Computer Science

Software Engineering

**Keywords**