Role of Functional Clones in Software Development

Kavitha Esther Rajakumari¹, Dr.S.Srinivasan²

¹Sathyabama University
²Anna University

Abstract: The software organizations have a great demand on producing quality software. The methodology used to produce quality software varies depending on the organization’s time, cost and resource constraints. In most cases, as the need for quality arises cost and time also increases. An efficient technique helps to cope up with these constraints and deliver a quality software system. In this paper, a cost-effective method is proposed to produce quality software. This proposed method uses the concept of code clones to effectively develop quality software. Most of the surveys regarding code clones project its negative aspects, but here the positive side of it is focused. The results show that this method uses limited resources and also user-friendly.

Keywords: Quality software, software development, code clones, functional clones.

1. Introduction

Software engineering has three major tasks embedded in it. They are software development, software reuse and software maintenance. Developing quality software helps in the steady growth of an organization. The techniques and methods used by an organization in software development determine its economical growth. Some techniques help in developing quality software but very expensive. Some methods are complex which are too difficult to implement. Some are simple but time consuming. Some techniques make use of more resources which indirectly leads to lack of resources for other tasks. So choosing an appropriate method is very important in developing software. Most of the organizations are equipped with customized tools which help in software development. Some organizations especially small-scale industries are not able to face the challenge of producing quality products. Industries which have very limited resources are often considered as small-scale industries. The proposed method paves way for these types of organizations to carry out the development process successfully.

In this paper, the proposed method makes use of code clones. Implementing this method is less expensive, affordable and meets the requirements of small-scale industries. Code clones are identical codes present in a software system. They are also known as duplicate codes. There are four types of code clones. Type-1 clones are the exact replica of the original code fragments. Type-2 clones are also identical to original fragments but names of variables, identifiers, etc. are changed. Type-3 clones are ones were the original fragment is modified i.e. additional changes or deletions are done. Type-4 clones are functionally similar. In general, code clones are considered harmful which deteriorate the software’s quality. But by careful analysis beneficial clones can be extracted which aid in improving the quality.

2. Literature Survey

The survey based on software engineering and code clones are discussed in this section. Francesca and his team mates [1] have discussed about the various tools assist in detecting code clones. They consider clones as bad smells. James R Cordy and Chanchal K. Roy [2] have proposed a tool named DebCheck to detect code clones in open source software systems. Dongxiang and Miryung [3] have found that clones that stay in the program over a long period of time cannot be refactored as soon as it is detected. They have termed those code clones as long-lived clones. The authors have mentioned that clones are not necessarily harmful.
The need for a new software development approach was discussed by Rupinder and Dr. Jyotsna [4]. They have listed some of the major issues faced in software development such as over budget, premature termination of projects etc. The concept of sharing knowledge among the development team members is stated as the key process in software development by Sharon and Rory [5]. Jesper and his peer group [6] have indicated that component reuse and iterative processes increase adaptability to changing customer needs.

The above discussions show that clones are identified as one of the bad smells present in the software. Therefore detecting and removing clones is one of the processes while developing software systems. Moreover the software development process should be fine tuned and must be able to overcome the drawbacks of the organization. Although the demerits of an organization have an impact on the products being developed, proper steps should be taken to maintain quality at an acceptable level.

3. Proposed Method

The proposed method makes use of the data mining approach to extract functional clones. Among the available data mining algorithms, FP-Growth algorithm [7, 8] is used. Though it is an existing algorithm, it proves to be efficient when used in source codes. The input to the system can be any software application. The output gives the set of functional clones used in the software system. They are maintained in the database for future use.

The proposed method consists of five steps, each of which is briefly given below.

• **Input Selection:** In this phase, the input to the system is selected. Mostly the real-time applications are taken so that harmful codes are less in number. Sometimes not-in-use applications are also taken as inputs. Not-in-use means old version application software which was once efficiently working but due to new version release has stopped functioning.

• **Module Separation:** Here the selected software application is separated into modules for further investigation. Each module becomes an input to the proposed system. The application as a whole is also possible to be given as an input but problem of segregating the detected functional clones evolves. As a result modules are taken as input so that classifying functional clones according to their usage will be a simple task.

• **Functional Clones Extraction:** After module separation, functional clones from each module are extracted. Apart from library functions, functions that have occurred more than ones are considered as clones. Beneficial functions with more than five occurrences can be made as library functions for future use.

• **Functional Clones Categorization:** In this phase the extracted functional clones are examined manually to check for harmful codes. If noxious codes are present within a function, those functions are removed. Based on the inspection of codes present inside a function, decision will be made either to reject or retain the function. Beneficial functions are retained and unproductive functions are discarded.

• **Storage of Beneficial Clones:** In this step, the clones that were termed beneficial in the previous phase are stored separately in a database. The database must be highly secured in order to avoid inconsistency among the stored data. The developers alone have access to the database apart from the system administrator and project manager. This database helps in providing useful functional clones in software development, software reuse and software maintenance.

Thus the proposed method assists in the software engineering process efficiently. It helps the developers to complete their project work within the stipulated period. This prospective achieves quality software development in an inexpensive means. So each time when a software is under development the programmer or the developer must keep in mind that while repeating certain codes or functions it should be reasonable and valuable to pave way for future usage. The concept of the proposed method is depicted below.
4. Implementation

An application software is taken as the input for the proposed system. It asks for the user name and password for authentication purpose so that unauthorized person cannot access the system. This is shown in Fig.2. In Fig.3 browsing the input software is displayed. After entering the system, can browse any application software stored in the database. Here the Project Management System is taken as input.

Fig. 2: Login Page
After the input software is given to the system, the application software is separated into modules. The Fig.4 below shows some of the modules and the relevant clones detected in each module. Each functional clones extracted from the modules are checked manually for their correctness. When each functional clone is clicked its relevant code is opened. Based on the codes, the usage of the clones are determined and categorized. This process is depicted in Fig.5.
After categorization, the details of detected clones are displayed. Number of occurrences, line number, page number and the module in which it occurs etc, are given. These details provide the exact location of the extracted clones. These details are a great help when need to track the origin of clones in the future. This conceptualization is shown in Fig.6.

![Fig. 6: Functional clones description](image)

The figures 7 & 8 displayed below shows the software development process of inserting functional clones in a software under development wherever necessary depending on their usage.

![Fig. 7: Insertion of a functional clone during s/w development](image)
The last two figures 9 & 10 below exhibits the database where the beneficial clones are stored and maintained. Both the databases support the developer during software development.
5. Conclusion and Future Work

Code clones have their own advantages and disadvantages. But by careful analysis of detected clones, the beneficial clones are used for software development. Therefore the proposed method proves to be cost-effective and most likely to be used in small-scale industries. This method is user friendly, easy to implement and easy to configure depending on the requirements of the organization.

In the future this method can be extended by adding various features according to the needs of the organization.

6. References