Enhancing the Code Clone Detection Algorithm by using Biological Technique

A Dissertation Proposal Submitted

By

Jaspreet Kaur

(11304255)

To

Department of Science & Technology

In partial fulfillment of the Requirement for the

Award of the Degree of

Master of Technology in Computer Science

Under the guidance of

Gurpreet Singh

(May 2015)

LOVELY PROFESSIONAL UNIVERSITY

School of: Computer Science Engineering

DISSERTATION TOPIC APPROVAL PERFORMA

Name of the Student: Inderjeet Kaur Registration No: 11304255

Batch: 2013-15 Roll No: B36

Session: 2014-15 Parent Section: K2307

Details of Supervisor:

Name: Gurpreet Singh Designation: A.P

U.ID: 16523 Qualification: M.E

Research Experience: 3 yrs

SPECIALIZATION AREA: Software Engg. (pick from list of provided specialization areas by DAA)

PROPOSED TOPIC:

1. Analysing & Enhancing various code clone detection Techniques
2. Enhancement in Black Box Testing
3. Enhancement in Selenium tool for Selenium Testing

Signature of Supervisor 16523

PAC Remarks:

Topic (1) is approved, Publication Expected.

APPROVAL OF PAC CHAIRPERSON: Signature: Date: 24/9/14

*Supervisors should finally encircle one topic out of three proposed topics and put up for approval before Project Aparoval Committee (PAC)

ABSTRACT

The means of software reuse is copying and modifying block of code that result in cloning. According to survey, it is observed that 20-30% of module in system contain clone. Mostly, peoples copy the work of another because they are not able to start the work from scratch and don't have time to complete it. Due to this lack of knowledge, the more clones are generated in software, that increase maintenance cost. So it is mandatory to detect clones in system to reduce replication and improve reusability.

The copying the existing code and paste them with or without modifications into other section is known as a software cloning. The copied code is called software clone.

Code clone is similar or duplicate code in source code that is created either by replication or some modifications. Clone is a persistent form of software reuse that affects the maintenance of large software.

In previous research, the researcher emphasized on detection of type 1, type 2, and type 3 clones. The existing code clone detection tools are used to detect clone in source code. In this research, the enhancement in code clone detection algorithm will be proposed which detect type 4. In this work, firstly, an existing algorithm is used to detect clone. Secondly, put some intensification in that algorithm to detect clone. Thirdly, combine algorithm with type 4 to detect a clone in particular function.

By using type 4, the efficiency of clone detection is increased. Clone is detected in particular function, which is more accurate and more efficient in manner.

ACKNOWLEDGEMENT

I would like to express my special thanks to God to give me this opportunity of writing this thesis and providing such nice peoples who was there always to help me in my dissertation. Secondly, big thanks goes to my mentor “Gurpreet Singh” who gave me this topic “Code Clone Detection” for my dissertation work, I am heartily thankful to Gurpreet Sir for being my mentor and helped me in doing a lot of Research and I came to know about so many new things. Very special thanks to all the authors whose paper I referred for this dissertation. Their effort made me to think about new ideas and due to what I am able to implement them in my research. “Source: Internet” gave me so many short definitions that’s included here.

At last, I would also like to thank my parents and friends who helped me a lot in my research within the limited time frame.

DECLARATION

I am **Jaspreet Kaur** hereby declare that the dissertation proposal entitled "Enhancing the Code Clone Detection Algorithm by using Biological technique**"** submitted for the M.Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date: 01-05-2015 Investigator

Reg. No. 11304255

CERTIFICATE

This is to certify that Jaspreet Kaur has completed M.Tech dissertation proposal titled "**Enhancing Code Clone Detection Algorithm by using Biological Technique**" under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of the dissertation proposal has ever been submitted for any other degree or diploma.

The dissertation proposal is fit for the submission and the partial fulfillment of the conditions for the award of M.Tech Computer Science & Engineering.

Date: 01-05-2015 Signature of Advisor

Name: **Gurpreet Singh**

UID: 16523

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

Software engineering is about building, evolving and maintaining software systems. Software engineering is a set of problem solving skills, techniques, technology and methods applied upon a variety of domains to evolve and create useful systems that solve many problems like practical problems. Software engineer is required to handle software engineering projects which discover, create, build software and tells its behavior [1].Software is a non-tangible device like documentation and computer programs and it is different from tangible hardware device. Software Engineering is the discipline of computer science which applies engineering principles to create, operate, modify and maintain of software components [24]. An organized and systematically approach is adopted by software engineers regarding their work using some techniques and tools depending upon the resources available and problem to be solved. System engineering is different from software engineering. System engineering is concern with deployment, architectural design and integration where as software engineering is concern with development, quality and testing and control of the system [2].

The main goals of software engineering are as follows.

- To produce software of high quality having less cost.
- To achieve Correctability.
- To gain reliability.
- To improve efficiency.
- To produce the system under budget and on schedule.

1.1 Basic Activities of Software Engineering [3]:

The basic activities which are necessary to follow in software engineering are as follow:

- Define software process which is to be used.
- Manage development of project.
- Describing software projects which are going to use.
- Design of the product
- Implementation of the product
- Testing of the product.
- Integrates sub parts and test them as a whole.
- Maintance of the system.

1.2 Software Engineering Sub-disciplines:

There are many sub-disciplines of software engineering which are as follow:

1.2.1 Software Requirement:

A requirement specification is the complete description for the behavior of the system**.** It may be defined according to the system specification.

1.2.2 Software Design

It is a problem solving process and plan for solutions. Dataflow Diagrams and Flowcharts, are comes under software design. In this SRS document are transform into design form using some tools.

1.2.3 Software Construction

It is a formation, functioning which is in depth manner and in the construction process we can define methods of the process and its description. It helps to improve software quality.

1.2.4 Software Testing

It checks whether the expected results match with the actual results. This process is to recognize correctness, integrity and effectiveness of computer software.

1.2.5 Software Maintenance

It is the alteration of the software products for their correction after the delivery to correct faults, error and bugs in software. It is a changing of software after delivering to the customer.

1.2.6 Software Configuration Management

It provides the auditing, changes and report to the changes that are made. So we can say that SCM is a change management.

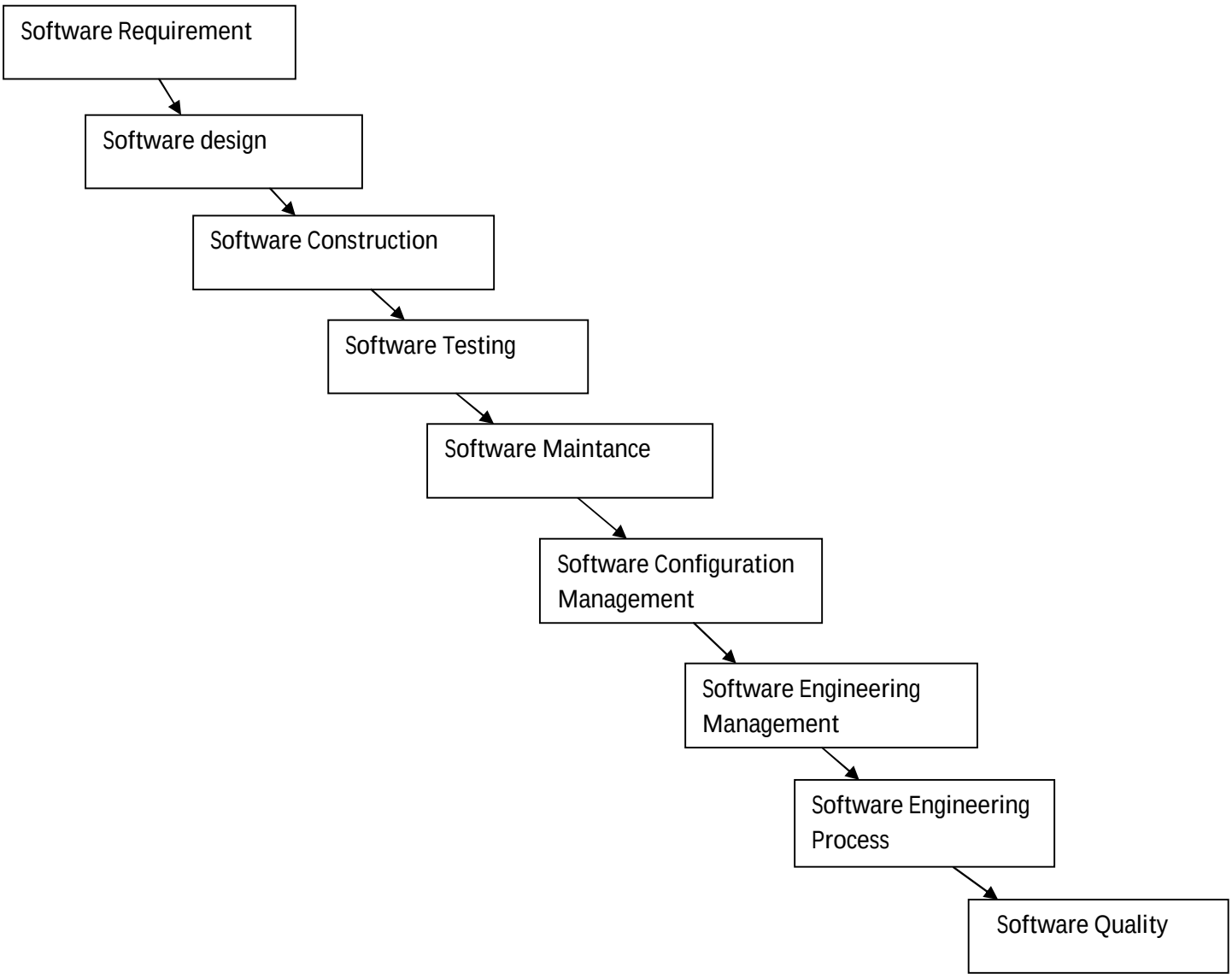


Fig. 1.1: Disciplines of Software Engineering

1.3 Software Testing

This process is to find out error or faults in a system to make it correctness, completeness and to identify the quality of already developed software (guru 99). This process is used to find the bugs and uncover it. Software testing is a process and discipline also. It is different from software development. It should be considered that is part of software development .It is an internal part of software development and closely related to software quality. The main aim of software testing is to fulfill user's requirements and make the system error free. So software testing is mainly to find outs the error or bugs to improve the quality of the system [2]. It is the last phase of the product before deliver to the customer. Software testing is an important part which tells whether the product is efficient and error-free, work properly and according to the requirements of the customer.

1.3.1 Software Testing Terminologies

The basic terminologies are as follows:

- **Error:** A mistake in coding is known as error. It is a misunderstanding of the internal stage.
- **Defect:** It is detected by the tester. It is some trouble in internal and external behavior of the products.
- **Bug:** the defect which is accepted by the developer and does not fulfill users requirement
- **Fault:** The invalid step taken which cause problem in future and give improper results.
- **Failure:** the result of fault is known as failure. The system is unable to perform according to the given specification.

1.3.2 Objective of Software Testing

The main objective of software testing is as follows:

- To find outs bugs or errors in a system.

- To improve the quality of a system and gain confidence.
- To prevent the system from the errors or bugs.
- Having less cost and more efficient system.

1.3.3 Essential Elements for Software Testing

To improve the efficiency and effectiveness of software there are five elements which are essential for software testing. These elements are as following.

- Test Strategy
- Test Plan
- Test Cases
- Test data

.

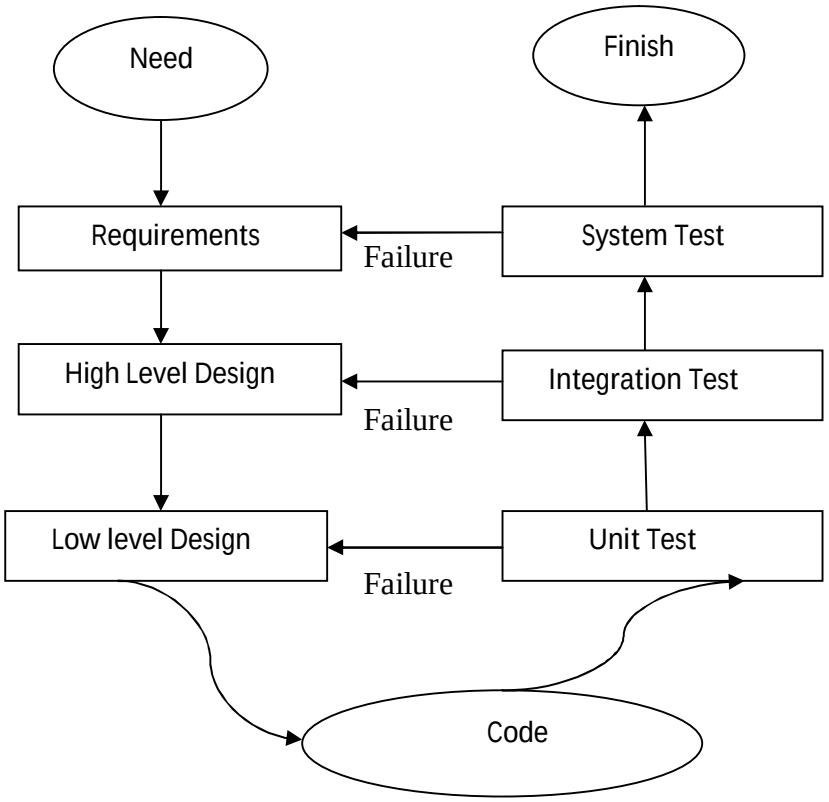


Fig 2:- Complete Cycle for Testing

1.3.4Types of Testing

There are different types of software testing are available to find out the defects in a software. These testing are as follow:

1.3.4.1 White-Box Testing

This testing also called as Glass testing and structural testing. In this, testing code is visible. It has knowledge of the internal mechanism of the components [18]. White-box testers are aware about the internal structure and also know how code is looks like. It is used in the validation process. It is a clear box testing because code can be easily visible in this type of testing.

1.3.4.2. Black-Box Testing

It is also known as functional testing. This testing ignores the internal mechanism of the system. It is a testing which is based upon the output and having no knowledge of internal code [1].It is attesting in which its working is not understood by its user. It has no knowledge of processing of code but only concentrate upon the output. It is used in the validation process. It is based upon the requirements and functionality. There is no user requirement in this type of testing.

1.3.4.3 Unit Testing

It is a type of white-box testing. It is a testing for low-level design code. It is done within a class and starts from a individual module [25].It has testing the smallest unit of elements of a software which module or component or unit .It is basically done by programmer not tester and require detail of structure of code. It helps to design test driver.

1.3.4.4. Integration Testing

This testing is done when two or more modules are combined together into a larger module. It verifies the functionality of the module after integration. It is done at the interfaces of the both the structure and component module. This type of testing is done in distributed or client/server modules. It uses both white box and black-box techniques [25].

1.3.4.5. System Testing

It is also known as end-to-end testing. It tests the complete application of environments. It is based upon the specification and requirement of the system. It checks the entire systems. It is for high level design and comes under the black-box testing [25].It also checks non-functional requirements also.

1.3.4.5.1 General Classification of Software Testing

There are mainly two types of testing:- Black Box Testing and White Box Testing. The diagrammatically representation of testing is as follows:-

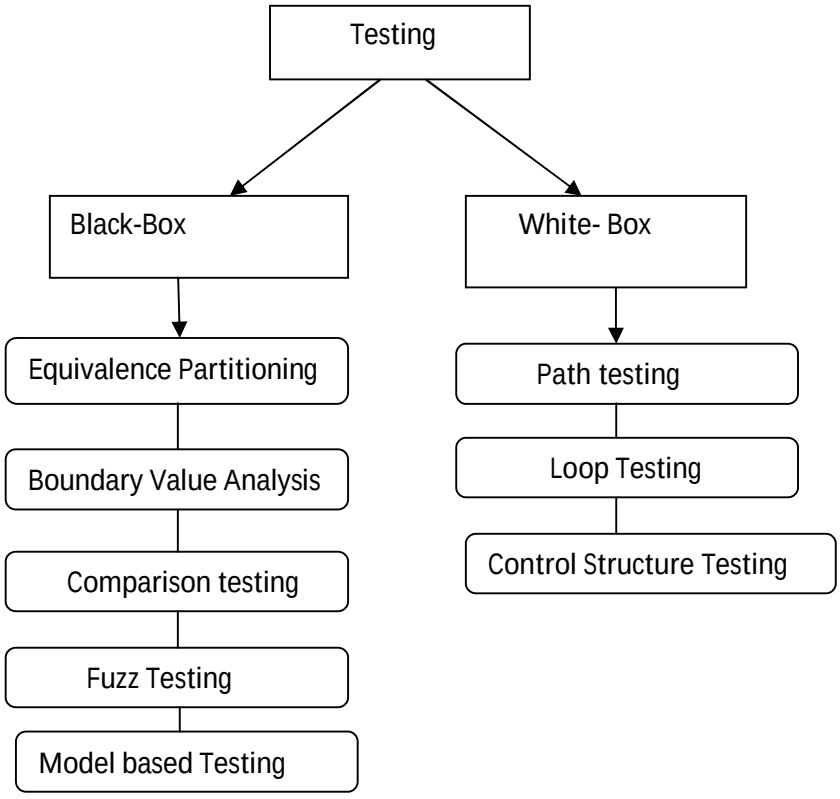


Fig: - General Classification of Software Testing

1.3.5 Approaches of Software Testing

There are three types of approaches which are followed by the software testing. These approaches are as following:

- Bottom-Up Approach
- Top-Down Approach

1.3.5.1. Bottom-Up Approach

Bottom–up is a one of the approach of software testing approach. It comes under integration testing. It starts with the unit modules such as programs and module and complete until the final phases reach. The control flow moves to upward direction. Drivers are used in the bottom up approach in the unit module. In this approach testing is conducted from sub module to main module. Drivers are used to simulate the main module. It is a temporary program. It is also known inductive reasoning.

1.3.5.1.1 Advantages of Bottom-Up Approach:

The main advantages of Bottom-up Approach are as following:

- Test results are easily observable.
- Easily test cases are generated.
- Faults are occurring at the bottom of the module only.

1.3.5.1.2 Disadvantages of Bottom-Up Approach

Disadvantages of bottom-up approach are as following:

- Driver modules are required.
- After the addition of last module only than program can exist.

1.3.5.2. Top-Down Approach

It is the second type of testing approach. In this approach we start from top level to bottom level. The control flow moves from top level to lower levels.tn this process module is created first then broken down into sub modules. Then sub module are further tested and broken down into super sub modules. Stub is used in it as a temporary program [21].

1.3.5.2.1. Advantages of Top-Down Approach

The main advantages of Top-down Approach are as following:

- Test cases are easily created after the I/O functions are added.
- If faults are find out at the early stage then it is useful.

1.3.5.2.2Disadvantages of Bottom-Up Approach

Disadvantages of Top-Down Approach as follow:

- Stub is complicated to produce at early stage.
- Observation of test output is difficult to produce.
- Testing and design are overlapped.

1.4 Clone Testing:

Software engineering is a method to develop, design operation and maintenance of software. Consequently in software engineering main focus is on to assure the quality in the product, detect the bugs and prevent system from bugs by testing or analysis. While developing any software for saving time and effort, software developer copy paste program code again and again. So if one module has bug, it is reproduced in every copy. There are several copies of code present and no record of such copies is present. This will make hard to fix such bugs and maintenance of existing software.

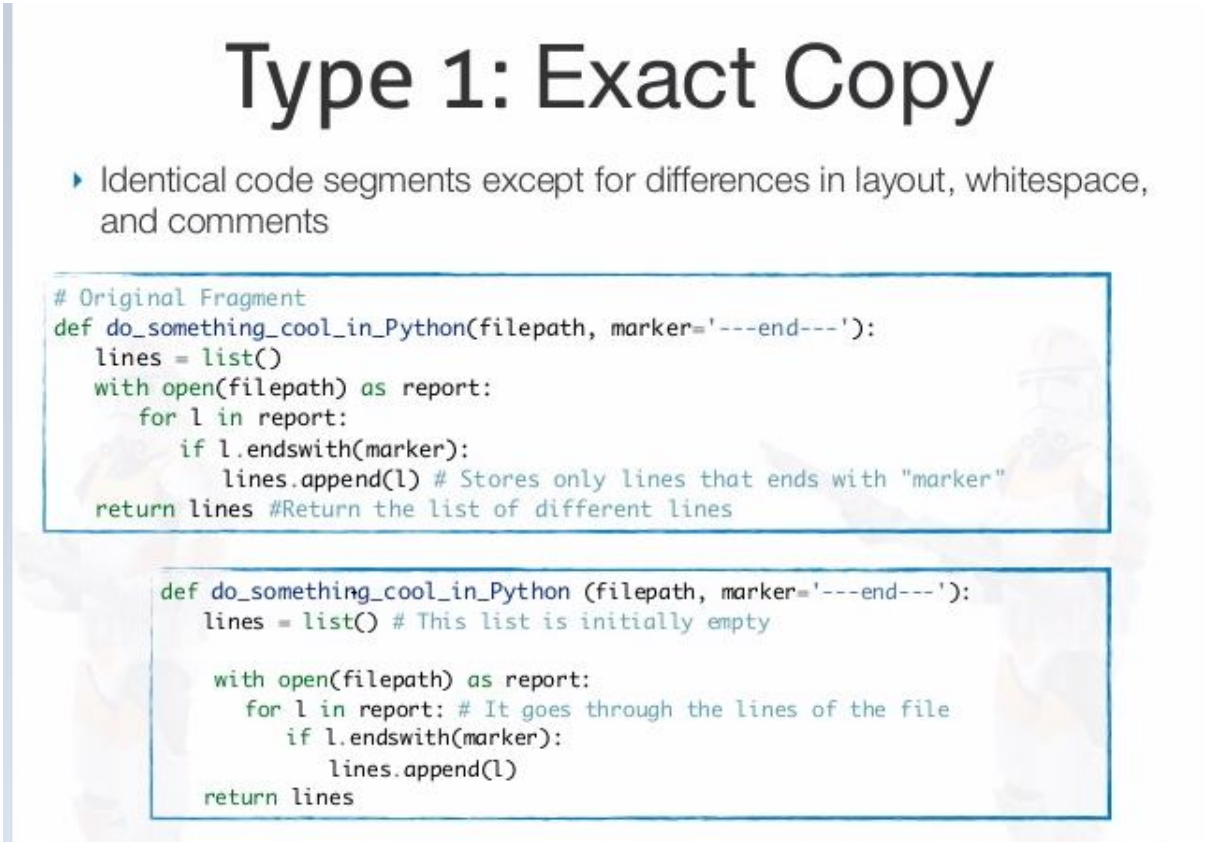
By concluded that the clone result comes from adding some extra functionality, which is same but not identical to existing logic. A clone in code is nothing but a similar or duplicate code in a source code or created either by replication or some modifications [3].These cloned code add to high maintenance cost of software and also cause the code bloating. This is because when changes perform on one clone, then the same action is performed on respected clone, this will increase the maintenance. These clones can also increase risk of faults in system [3]. Past research conclude that around 8%-25%, the source code in a software system contains code clone [3].The number of tools which are used to detect the code clones, but it is not effective to

remove the clones. So we can apply the principal of refactoring or modularity to improve the reusability and maintainability of software from clone code.

Types of code clones:-

The clones are divided into four types:-

Type 1:-These code clones are identical code clones that only allow modification in white space and comments.



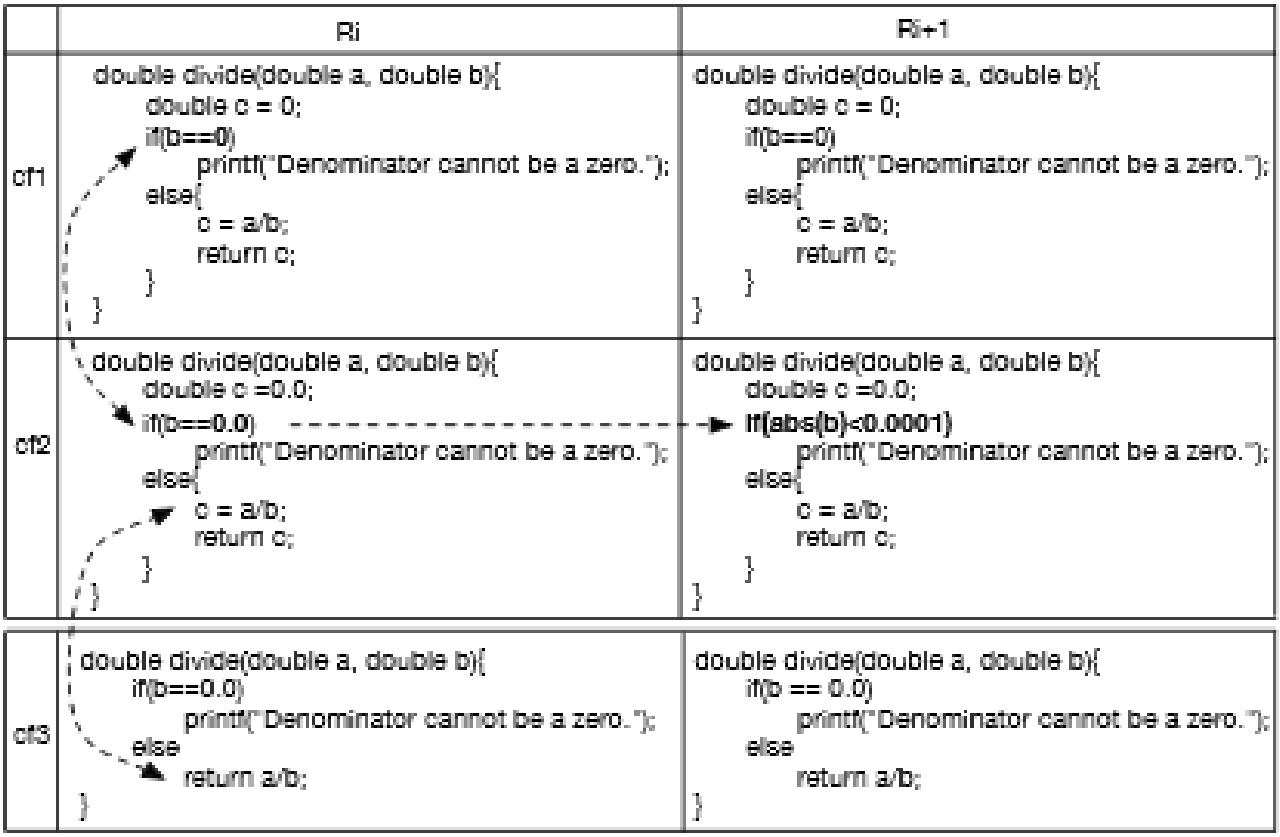
Type 2:-These code clones are semantically and syntactically identical copies.

- similar code fragments (syntactically or semantically)

```
static PyObject *
float_add(PyObject *v, PyObject *w)
{
   double a,b;
   CONVERT_TO_DOUBLE(v,a);
   CONVERT_TO_DOUBLE(w,b);
   PyFPE_START_PROTECT("add",return 0)
   a = a + b;
   PyFPE_END_PROTECT(a)
   return PyFloat_FromDouble(a);
}
```

```
static PyObject *
float_mul(PyObject *v, PyObject *w)
{
   double a,b;
   CONVERT_TO_DOUBLE(v,a);
   CONVERT_TO_DOUBLE(w,b);
   PyFPE_START_PROTECT("multiply",return 0)
   a = a * b;
   PyFPE_END_PROTECT(a)
   return PyFloat_FromDouble(a);
}
```

Type 3:-These code clones are copied fragments by changing, adding or removing statements.



Type 4:-These code clones are based on function similarity but they are different in syntax.

Techniques: - There are basically 4 types, that are:- textual, lexical, syntactic and semantic.

Textual approach: - In Textual approaches there is little need of normalization or transformation of code. In this, basically line to line comparison is done, which basically based on two types, one is simple line matching and other one is parameterized line matching. This technique is basically string based.

Lexical approach: - In lexical technique we convert source code into tokens using lexical rules. These tokens are then compared.

Syntactic approach: - In syntactic technique an abstract tree is generated. Using parser source code is converted into parse tree. Abstract tree is then processed either using tree matching or metric to find the clones.

Semantic approach: - In this approach, an source code is represented as program dependency graph. Nodes represent the statements and expressions and , edges represent control and data dependencies.

Advantages of clones

- Cloning helps to configure multiple software at once.
- It is an easy way to develop new software with the help of existing software.
- It is an only way to enhance the existing functionality.
- It is a fast and immediate method to fulfill the requirement of users.

Disadvantages of clones

- The presence of clone in software greatly increase the maintenance cost.
- Code cloning increases the probability of bug propagation. if any code contain a bug and that code pasted at different places, the same bug will be present in all the code fragments.
- When putting a strain on a system, then code cloning increase the size of software system.

Chapter 2

Review of Literature

C. K.Roy proposed a paper **"detection and analysis of near miss clones"[4]**. In this paper, A hybrid approach called NICAD is proposed to detect both exact and near miss clones. In this, a hybrid detection method is developed and then provides a scenario to compare and evaluate detection techniques and finally erect a framework for assessing clone detection tool.

NICAD works in three phases:-

- Standard pretty printing and extraction.
- Clustering and comparison.
- Reporting.

J.H.Johnson proposed a paper **"Identifying Redundancy in Source code using Fimgerprints"[5]**. In this paper, a large program source tree over a 300 megabytes has more clones and that exact duplication of text is detected by fingerprints. This approach is very suitable for envisioning and understanding the program. In other words, the information present in source is achieved by looking for repeated substrings.

B. S.Baker presented a paper **"On Finding Duplication and Near Duplication in Large Software System"[6]**. In this paper a program DUP is used to identify the code which is copied and modified. DUP ignore the indentation in any case. By using DUP, firstly fix the bugs,then match the same bugs that are present in code, then finding the identical and near identical bugs present in the code.

T. Kamiya proposed a paper **"CC Finder: A Multilinguistic Token based Code Clone Detection system for large scale source code"[7].**In this paper,a CC Finder tool is used to detect clone in software. This tool is used to detect code clone in c++,c,java,cobol and other

source file. As a output, CC finder detect and resolve clones in software. The main advantage of CC Finder is to raise performance and efficiency. This paper showed some issues that are generated in clone detection. The issues are:-

- Regularization of identifiers
- Measuring clones
- Identification if structure

Kodhai.E proposed a paper **"Clone Detection uses Textual and Metrics Analysis to Figure out All Types of Clone"[8]**. In this paper, each line is compared textually using hashing technique. This paper uses a metrics based technique to detect a functional clone. This paper mainly detect the code line by line. The conclusion of this paper is to implement various metrics technique to reduce the total comparison overhead.

R. sivakumar and Kodhari.E proposed a paper **"code clone detection in websites using hybrid approach"[9]**. This paper detect all types of clone in web application by using hybrid approach by the combination of textual and metrix analysis.this method is least complex and provide efficient way of clone detection. The conclusion of this paper is to find functional clone and eliminate the duplicate code in web application and improve the poorly designed web applications.

S. K.Abd-El-Hafiz published a paper **"A Metrix Based Data Mining Approach For Software Clone Detection"[10]**. This paper use data mining approach to detect a clone from software. In this fractal clustering algorithm is used. By using clustering algorithm, firstly metrics are collected form functions of software and then partition into clusters then clones are deterct from that clusters. This paper mainly focus to detect clone from clusters rather than line by line comparison to give accurate and less complex result.

A. Kaur, B. Singh published a paper **"Study on Metrix Based Approach for Detecting Software Code Clones"[11].** this paper used metric approach to detect the bugs directly from the software rather than working on source code. The principal of refractoring and modularity is applied to improve the maintainability of software. After the study of this

paper, we can conclude that the metrix based approach detect all types of clone form the software.

R. Koschke, R. Falke, P. Frenzel published a paper **" Clone Detection using Abstract Syntax Suffix Trees"[12].** which able to find syntactic clone by using Bellon benchmark technique. The token based clone detection based on the suffix tree. In this paper, firstly they use the first letter for the word,then construct the suffix tree, then identify the clone in the suffix tree. At the end they conclude that bellon benchmark technique is use to count token instead of line as a measure of clone size.

P. Bulychev and M. Minea presented a paper **"duplicate code detection using anti unification"[13]**. In this paper only syntactic similarities are considered. The main aim of this paper is to detect a wide range of clones i.e. third type of clone. In this, clones are detected using anti-unification algorithm. Firstly, partition statements into clusters and then fine all pair of identical sequence of clusters and then check for similarities by using the anti-unification algorithm.

J. Krinke presented a paper **"Identifying Similar Code with Program Dependency Graph"[14]**, this paper detect the similar code in directed graph. Such approach detect the duplicate code. Such approach is based on semantic, text and syntax. This approach is feasible in non-polynomial complexity of the problem.

R. komondoor and **Susan Horwitz** proposed a paper "**Using Slicing to Identify Duplication in Source Code"[15],** This paper describe the design and initial implementation of tool by using program dependency graph and program slicing to find isomorphic program dependency graph that detect clones. The benefit of this approach is to find the clones that does not occur component as a contiguous text. The main approach of this paper is to describe the implementation tools that find the duplicate code in C program and display to the programmer.

Rowyda Mohammad et.al have proposed "**A technique to extract exact clones from object oriented source code using differential file comparison[16]**.Objective of this technique is to detect cloned code ,which are reason of crosscutting concerns that leads to

reducing the system reusability and maintainability. DIFF(Differential File Comparison Algorithm) will find the different lines of code between two source codes files and rest of the lines of code are same and that will be considered as clones. Then that clones will extracted from code .This technique will pass through three stages: (1)Source code normalization (2)Differential File Comparison (3)Extracting Exact Clones.In the first stage little transformation of code is needed. During transformation ,the white spaces and comments are removed from code.In second stage DIFF determine the difference of lines between two files that are not changed. Unchanged lines will be declared as clones. It will solve the problem of longest common subsequences by maximize the size of lines which are unchanged. In third step it will Extract exact clones from two given source codes. Lines which are not different are considered as clones and which results in extracting the clones two source files. However ,this tool is implemented on C# language. This technique is implemented on only two source files. Although this technique is simple and take less time but it can only find type 1 clones .This tool can be extended to detect type 2 or type 3 clones and work on more than two source codes.

Saif Ur Rehman et.al (2012) has proposed **"A technique to detect code clones in large source codes"[17].** This technique will overcome key problem of other techniques to detect clones only in one language. This technique can detect clone in any size of source code.LSC Miner (Large Source Code) will detect the codes written in more than one language source code.LSC Miner is an prototype tool .In first step this technique will read the source code and forward it further to next stage. In second step it will take source code as input and then this code is converted into tokens. This technique is using two dimensional array for faster storing and identify the clones in source code. So these tokens are stored in this two dimensional array as dataset. In third step hash values are assigned to each converted token. Every hash has weight, which are the total number of token there in the statement. In last every hash value is compared and on that comparison result is out. Matched and copy paste data will be detected and considered as clones.LSC Miner is implemented in Visual Basic .Net 2.0 .This system found clones written in C ,C++,Java,VB.Net,C.

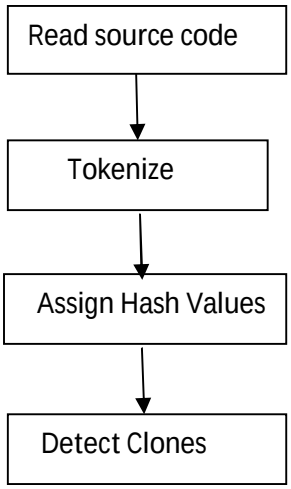


Figure 2: Flow of LSC Miner

Praveen Ranjan Srivastava *et al* (2011) has discussed **"Software testing Effort: An Accessment Through Fuzzy Criteria Approch"[18].** It consists of estimating testing effort, selecting appropriate test team, designing test cases, executing the software with those test cases and examining the results produced by those executions. It indicates cost of software development is committed to testing, with the percentage for testing critical software being even higher. This paper makes an attempt using fuzzy logic to estimate reliable software testing effort. In this paper triangular membership functions are chosen with monotonic constraints.

Sahil Batra *et al* (2011) have mentioned **"IMPROVING QUALITY USING TESTING STRATEGIES"[19].** In this paper, a variety of types of software testing technique and their different attributes of software quality are explained. The aim is to identify the types of testing that can be applied for checking a particular quality attribute. All types of testing cannot be useful in all phases of software development life cycle. Also summarized which types of testing are applicable in which phases of life cycle of software development. General SDLC processes are applied to different type of projects under different conditions and requirements. The debugging and testing differences are also explained.

Chapter 3

Present Work

3.1 Problem Formulation

Code cloning is the process of duplicating and modifying code, or creating replication of code fragments in the source code. Clone groups are type of code clones that are clones of each other .Clone Detection is a technique that is used to detect functions that are same of another function. The type of code clone is an indicator of complexity as well as the level of difficulty in identifying and detecting the clone. In this case, type 1 is the easiest to detect and type 4 the most difficult. Type 4 Cloning is used for the detection of clones in a particular function. The problem is that it makes clones of all the function it copies rather than only the copied lines. To overcome this problem we have to use the pattern matching algorithm so that only matched values are cloned, not other values.

Pattern Matching Algorithm is used to match the values that are cloned, but to match the values line by line, we are using Ant Colony Optimization Technique. This technique is inspired by foranging behavior of ant colony. Ant colony optimization takes inspiration from the real ants colonies and which are used to solve optimization problem.

3.2 Objectives

The main objectives of the study are as follow:

- To study all types of clone, type 1, type 2, type 3 and their algorithm.
- To identify the Type 4 cloned.
- To compare existing algorithm with proposing algorithm.
- To remove the problem of whole function cloned of type 4 by using patter matching algorithm.
- To reduces the complexity and improve the efficiency of the functions.

3.3 Research Methodology

- Explore all the existing algorithms to detect all types of clones.
- Apply type 4 enhancing in existing algorithm to detect function code clone.
- Reducing the complexity and improving reusability and maintainability by pattern matching complexity algorithm.

By removing this error we can use this for various purposes. After combining type 4 with pattern matching, complexity of the function is removed. In type 4 whole functions are cloned but with pattern matching algorithm few lines are cloned which is time saving process and fast than the existing function. In this case, the easiest detection type is type 1and most difficult type is type 4. Type 4 Cloning is used for the detection of clones in a particular function. But type 4 is difficult to because it make clone of all the function it is copied rather than only the copied lines. To overcome this problem we have to use the pattern matching algorithm so that only matched values are cloned not other values. So pattern matching algorithm helps to cloned the duplicate code only not the whole function.

Chapter 4

Result and Discussion

4.1 Introduction to MATLAB

MATLAB stands for matrix laboratory. It is a multi-prototype numerical computing environment and fourth generation programming language. It is used for matrix manipulation, plotting of function and data. With the help of its programming capabilities it provides tool which is very useful for all areas of science and engineering.

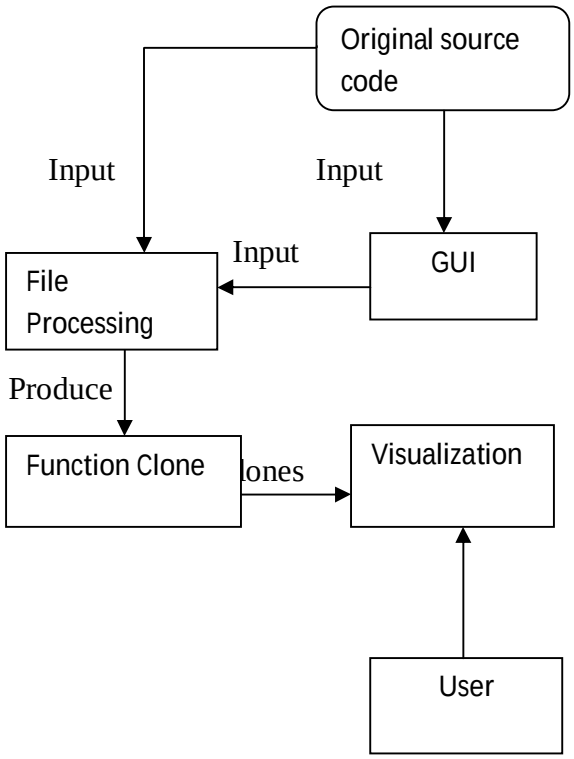
GUI toolbox allow advanced matlab programmer to provide graphical user interface to their program.

4.1.1 Introduction of Ant Colony Optimization Technique

Ant colony optimization is a technique to solve problems which is used to fine good path through graph.

- This technique is inspired by the foraging behavior of ant colony.
- Ant colony optimization takes inspiration from the real ants colonies and which are used to solve optimization problem.
- For e.g., While walking from food sources to the nest and vice versa, ants deposit pheromones on the ground, forming in this way a pheromone trail. Ants can smell the pheromone and they tend to choose, probabilistically, paths marked by strong pheromone concentrations.

4.1.2 Proposed Flow Diagram



4.1.3 Proposed ALGORITHM

Begin
Initialization: i:=1
while(i<=n)
if(Functions are distinct)
if(clone is distinct from any previously processed clone)
then
if(startNo_Clone has method parameters)
then
compose advice specification with parameter binding
Initialization: line:= startNo_Clone+ 1
while(line<= endNo_Clone)
copy line to Aspect text area
line++
end while
else
compose advice specification without parameter binding
Initialization: line:= startNo_Clone + 1

while(line<= endNo_Clone)
copy line to Aspect text area
line++
end while
end if
end if
i++
end while
Initialization: i=1
while(i<=n)
comment out clone from array full_File[ ][ ]
i++
end while
end begin

In this work, the enhancement is made in the existing algorithm, in existing algorithm only we can detects the clones in the whole code. In enhanced algorithm, the clone code under the specified function is also checked. As in the algorithm, if the function under which we are checking the cloned code has same name, that whole code is detected as the cloned code. If the function names are distinct, then the clone code detection process continues until whole code is been traversed. As experimental results shows that the enhanced algorithm is more efficient and detects the cloned code more efficiently in less amount of time.

4.2 Implementation

- **Tools for clone testing: -** The tools for clone will test both efficiency of new and old algorithm. In this figure, various checkboxes and buttons are used for testing the clone and results are shown in text boxes.

Fig 1 :- Clone Testing framework

As illustrated in the figure1, the interface is developed for clone testing. The tool will test both the efficiency of existing and new algorithm.

Results:- The outcomes that are generated from new algorithm and old algorithm are represented as follow. The below diagrams represent the efficiency of new algorithm, efficiency of old algorithm, and comparison of efficiency of new algorithm and old algorithm.

Efficiency of Old Algorithm:- In this, the clones are detected by using old algorithm. This old algorithm tells how much clones are detected in source code by using old algorithm.

Fig 2: Efficiency of Old algorithm

As illustrated in the figure 2, the first line of first code and two lines of the second code are checked and algorithm gave that 17.1115 % of the code are cloned with the old algorithm.

Graphical Representation of Efficiency of Old Algorithm:- This will show the efficiency of old algorithm in the form of graph. This represents

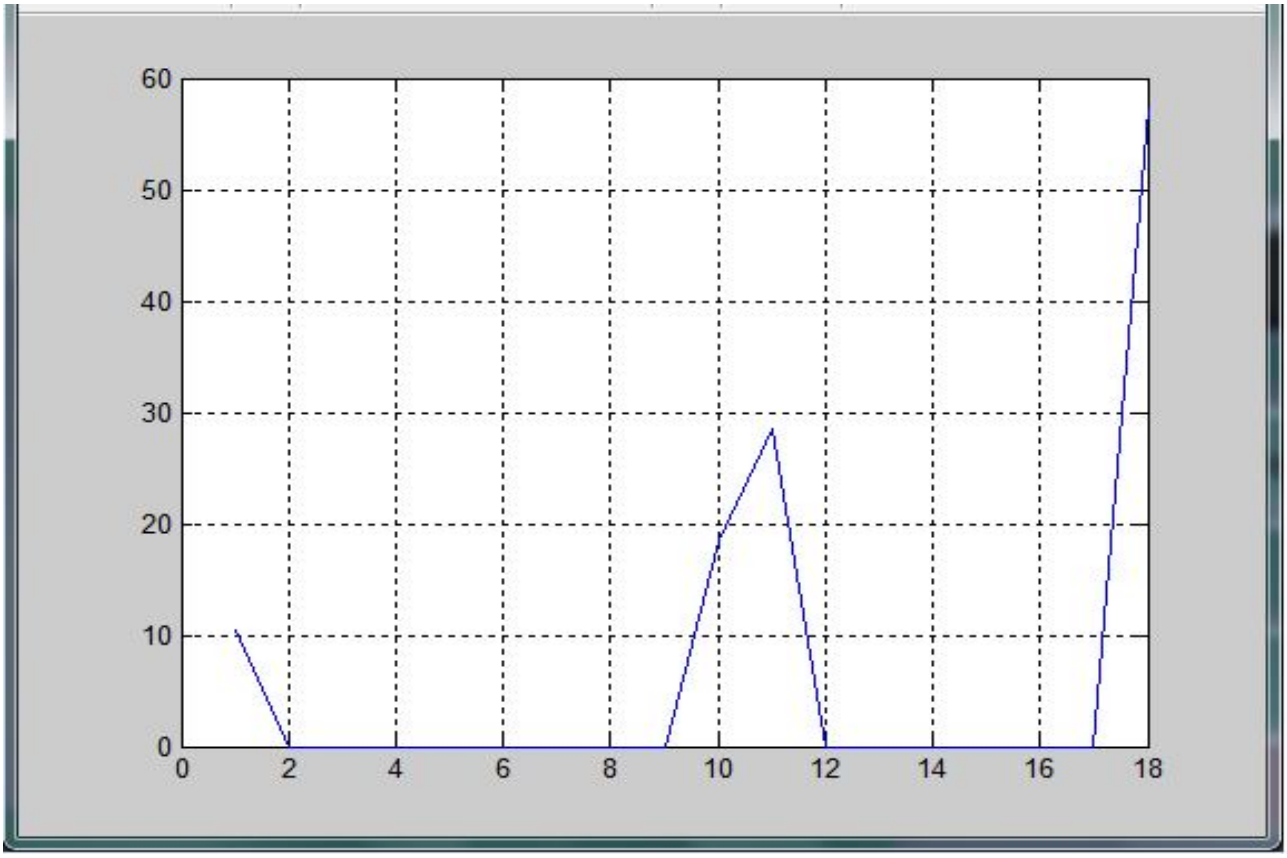


Fig 3: Efficiency of Old algorithm

As illustrated in the figure 3, the first line of first code and two lines of the second code are checked and algorithm gave that 17 % of the code are cloned with the old algorithm and correspond to their efficiency graph is been shown.

Bar Representation of Efficiency of Old Algorithm: - The area in this figure which is blue in color is cloned portion. It means that the clones are detected in this portion of source code. The result is generated from old algorithm. in this, the function clones are not detected.

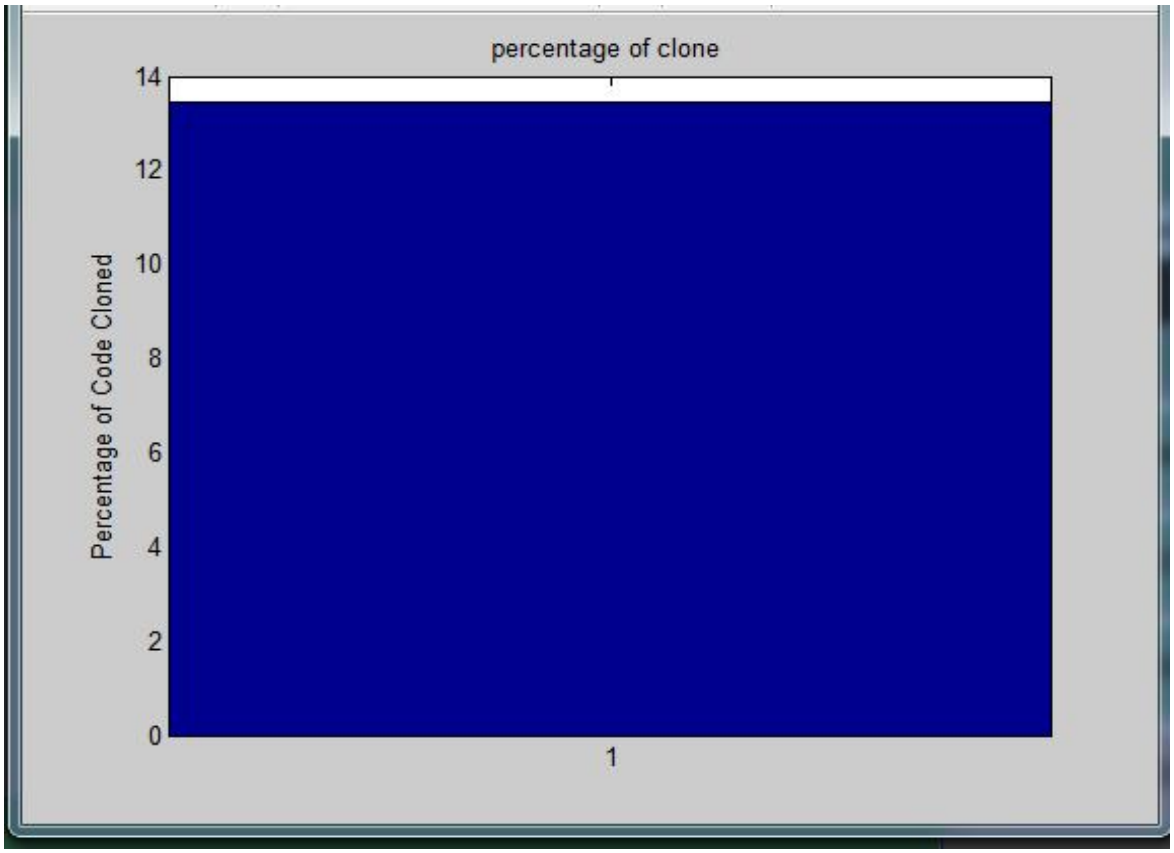


Fig 4:- Bar Representation of Efficiency of Old Algorithm

The above figure is the bar chart representation of the efficiency of old algorithm. In the above figure, the blue color represents the cloned portion. It means that the clones are detected in that portion of the code.

Efficiency of New Algorithm:- In this, the clones are detected by using proposed algorithm. This new algorithm tells how much function clones are detected in source code by using new algorithm.

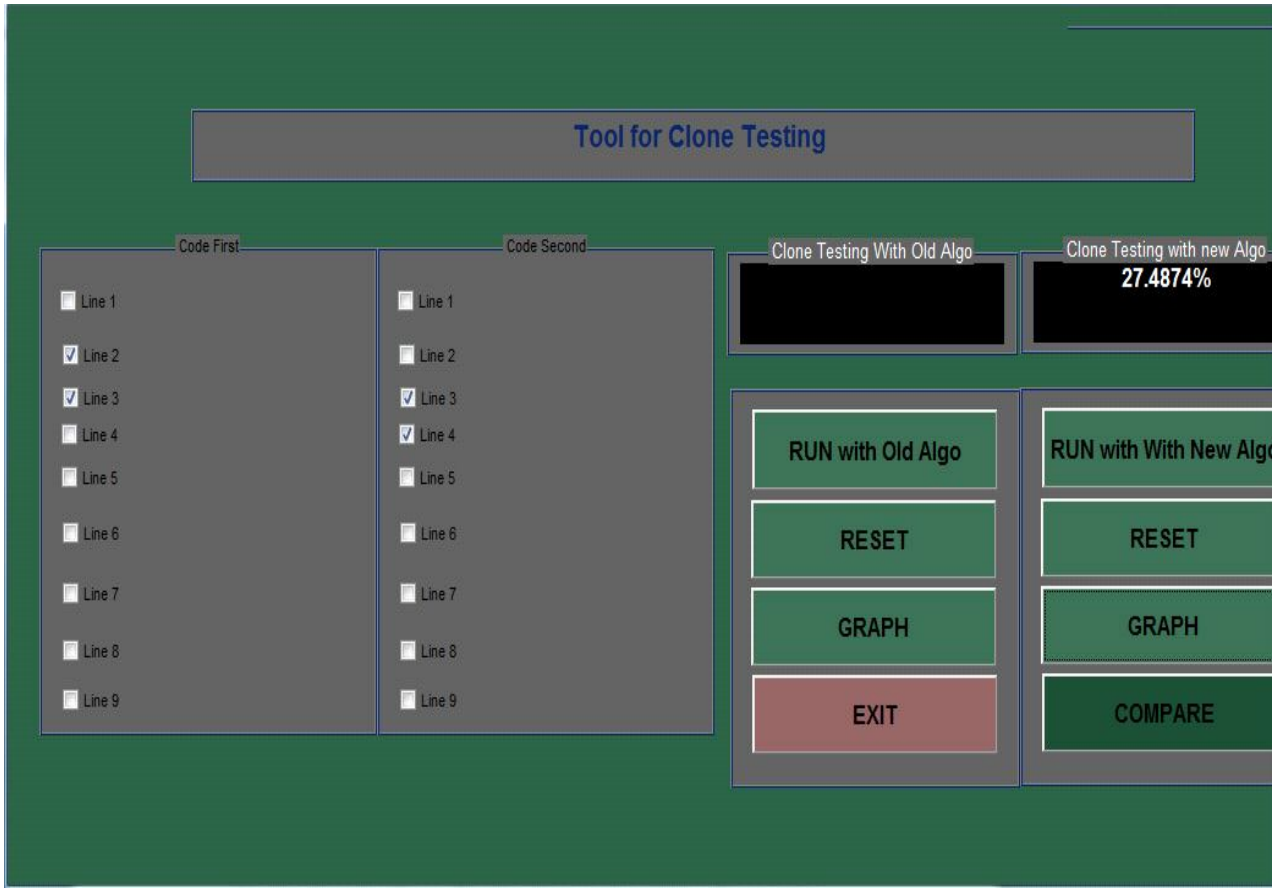


Fig 5: Efficiency of new algorithm

The above fig illustrate that the lines of first code and lines of the second code are checked and algorithm gave that 27.4874 % of the code are cloned with the new algorithm . this will increase the efficiency because it will also detect the function clones in source code.

Graphical Representation of Efficiency of New Algorithm:- This will show the efficiency of new algorithm in the form of graph. This represents

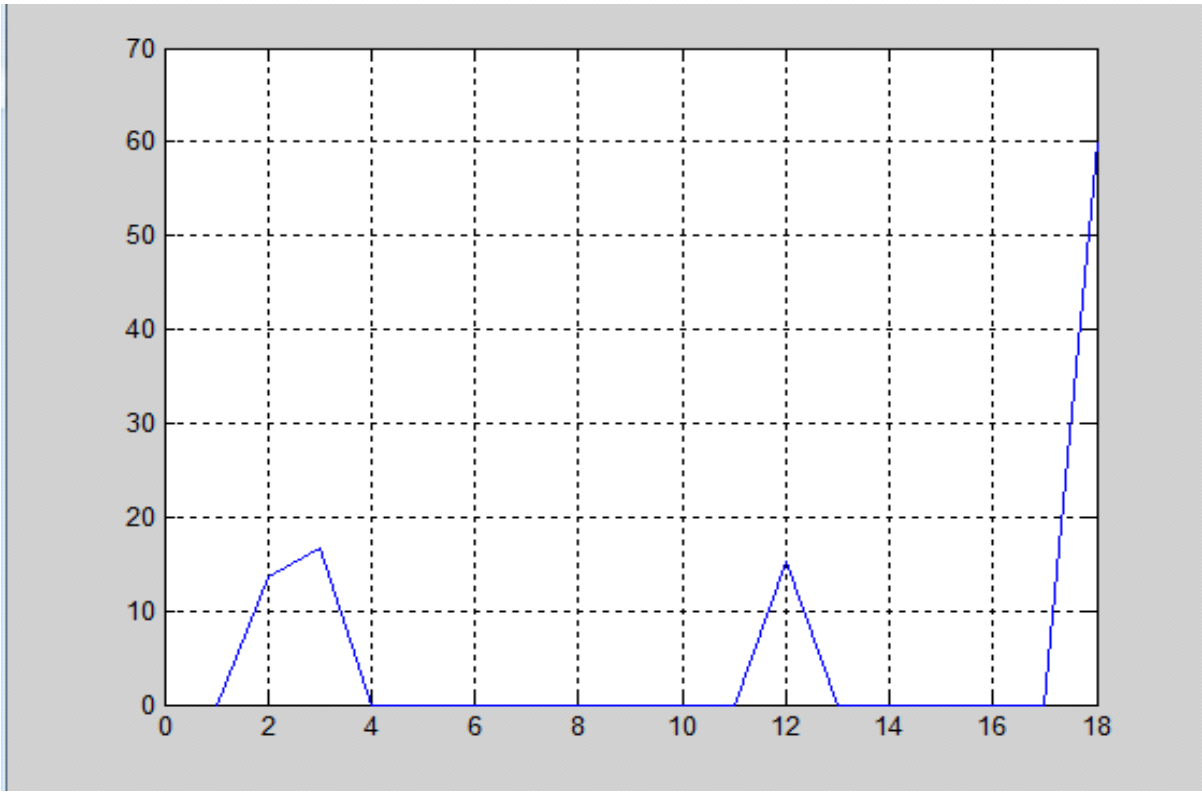


Fig 6: Efficiency of new algorithm

As illustrated in the figure 6 the first line of first code and two lines of the second code are checked and algorithm gave that 27.4874 % of the code are cloned with the old algorithm and correspond to their efficiency graph is been shown.

Bar Representation of Efficiency of New Algorithm: - The area in this figure which is blue in color is cloned portion. It means that the function clones are detected in this portion of source code. The result is generated from new algorithm.

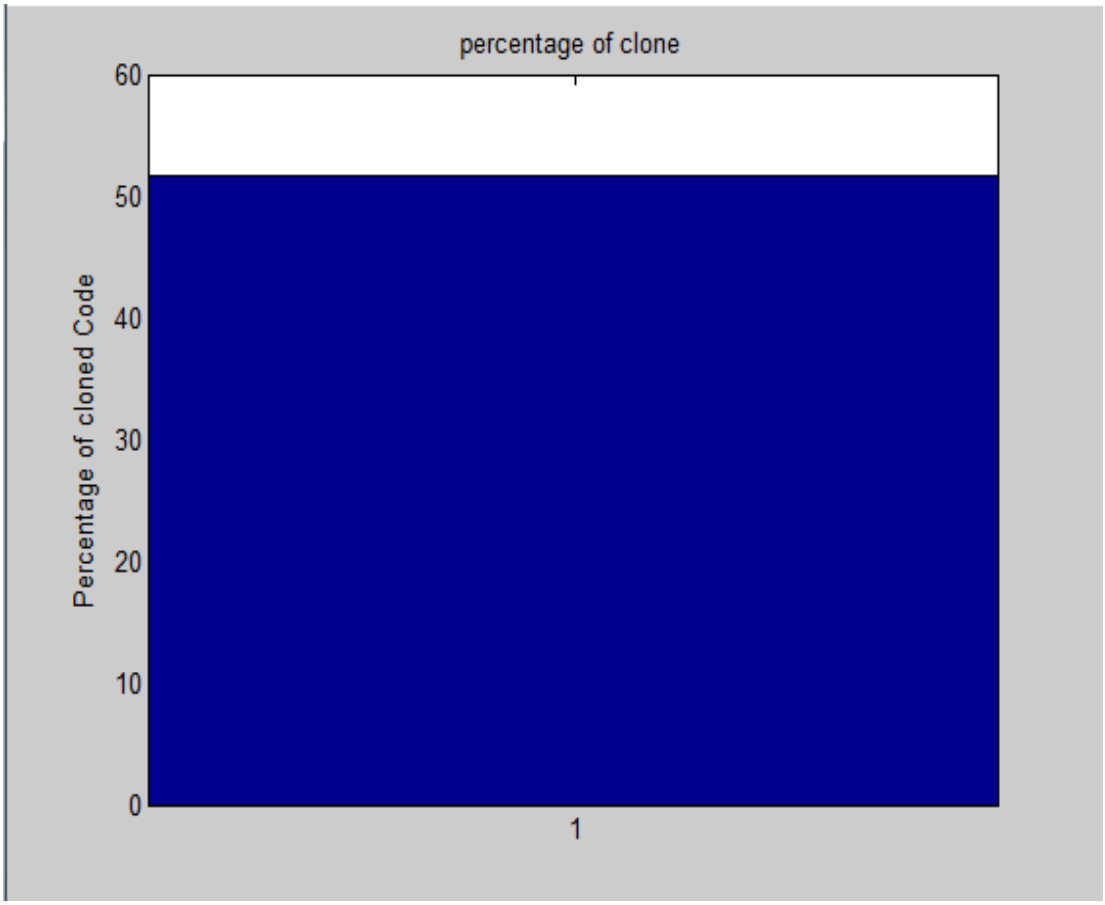


Fig 7:- Efficiency of New Algorithm

The above figure is the bar chart representation of the efficiency of new algorithm. In the above figure, the blue color represents the cloned portion. It means that the function clones are detected in that portion of the code.

Comparison of Old Algorithm with New Algorithm: - In the below figure, the clones are detected from old and new algorithm. The new algorithm is more efficient than new algorithm because it detects function clones.

Fig 8:- comparison of new algo with old algo

The fig 8 illustrates the comparison between new algorithm and old algorithm. The old algorithm detects only 22.9191% of clones, but new algorithm detects 27.4874% of clones. Because it detects code clone as well as function clones.

Graphical representation of comparison of new algo with old algo:-

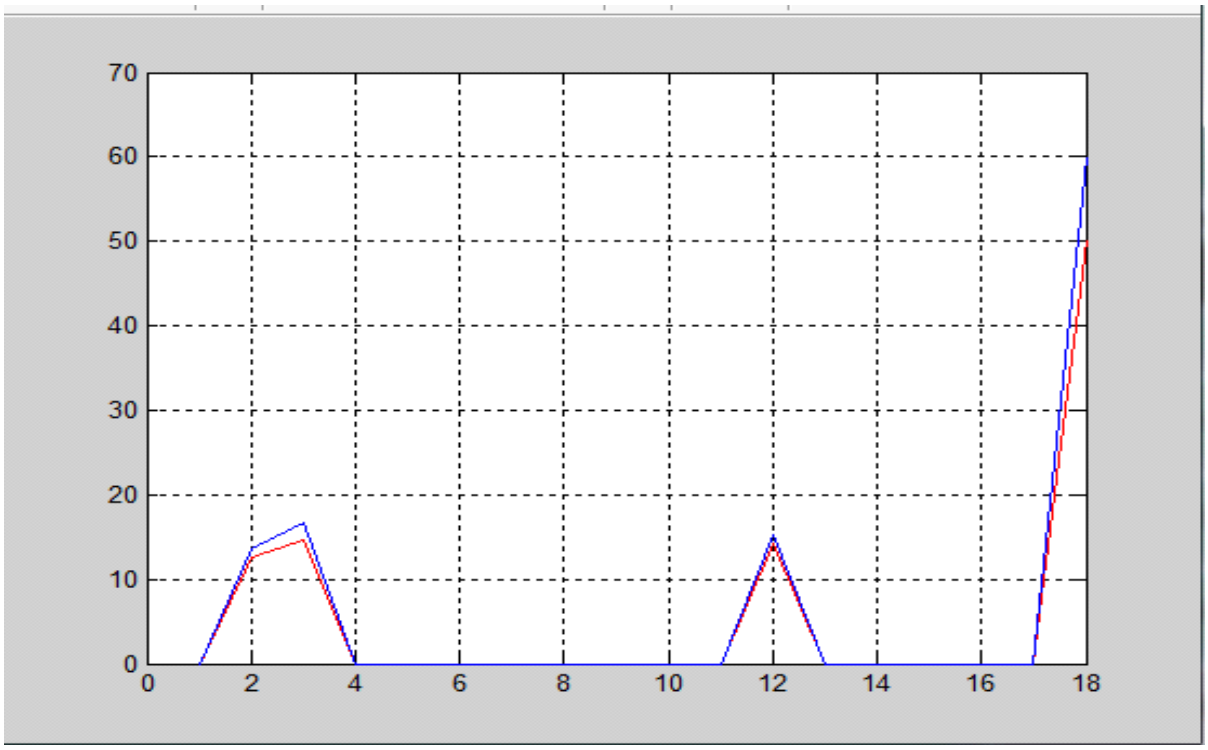


Fig 9:- comparison of new algo with old algo

As illustrated in the figure 9, the two lines of first code and two lines of the second code are checked and algorithm gave that 22.9191% of the code are cloned with the old algorithm and with the new algorithm it will be 27.4874%. and corresponding to their outputs graphs are shown

Bar Representation of Comparison of New Algorithm with Old Algorithm:-

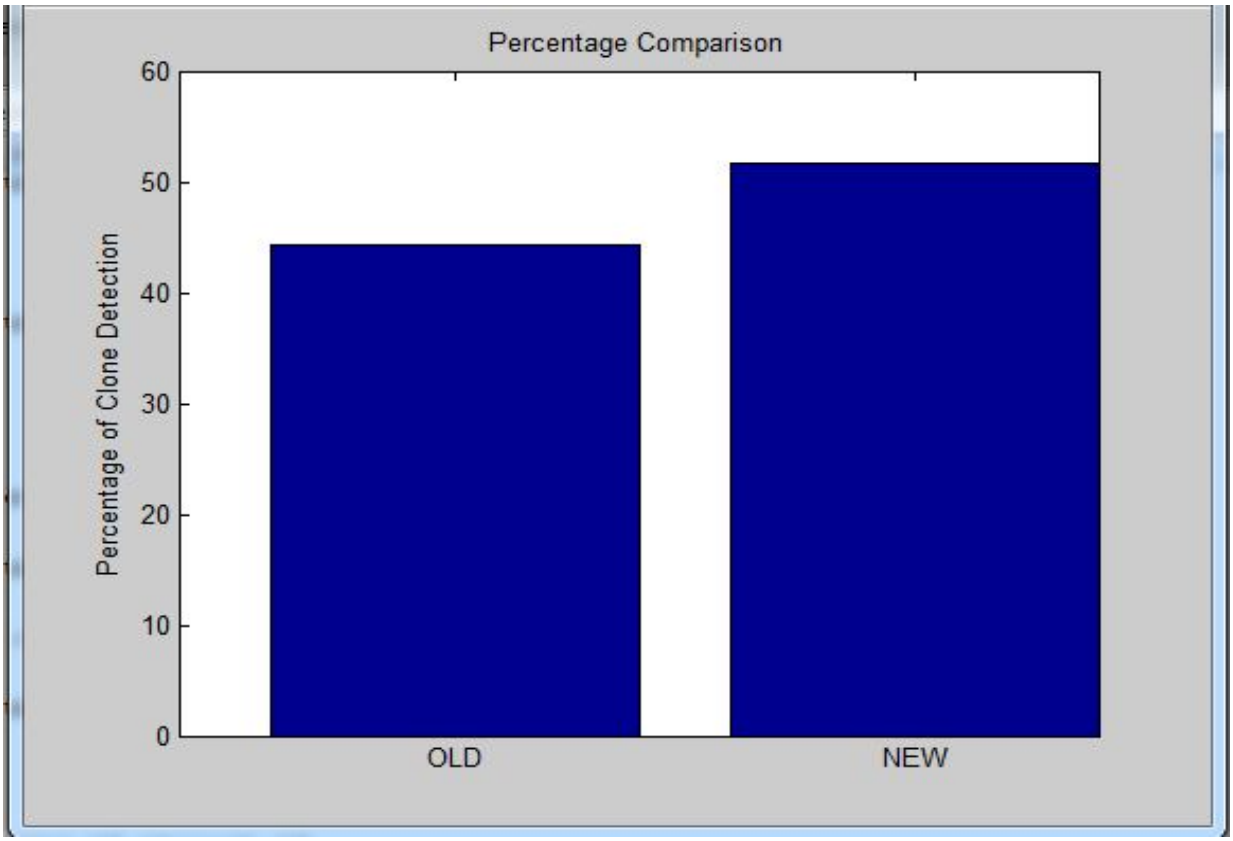


Fig 10:- comparison of new Algorithm with old Algorithm

The above figure is the bar chart representation of the comparison of new algorithm with old algorithm. In the above figure, the blue color represents the cloned portion. The efficiency of new algorithm is more than the efficiency of old algorithm.

Chapter 5

Conclusion and Future Scope

Software clone is a partition of code that is same according to some manners. Basically clone means duplicacy of source code. These clones affect the quality of system and more number of errors occurs in system.

This paper presented an approach that detects the clone in particular function. Basically code clone are very harmful to software system because they increase the maintenance cost. The approach in this paper can handle the type 4 of type of clone. This result shows that to detect the function clone in particular function increase the efficiency of clone detection.

The expected outcomes in this paper to detect the function clones in source code. This improves the efficiency in terms of execution time and accurate results. By using type 4 with pattern matching complexity algorithm, this approach improves the reusability and maintainability of software. By using type 4 with pattern matching algorithm and by using biological technique, this approach is more accurate and more efficient in manner than the previous techniques that are implementing to detect clones.

Future Scope: - In this work, the enhanced algorithm has proposed to detect function clones in source code. The further enhancement technique will detect clone line by line and will tell in which line, clone exist. It will also assign severity to the detected clones. The severity of the clones will provide better analysis in terms of code clone detection.

Chapter 6

List of References

I. References

[1] S.Biswas and Rajiv Mall , " An approach to software engineering", 2009.

[2] Myers, Glenford. (1979). The Art of Software Testing.

[3] J.Irena, "Software testing methods and techniques", 2002.

[4] C. K. Roy, "Detection and Analysis of Near-Miss Software Clones", Ph.D. Thesis, Queen's School of Computing.

[5] J.H. Johnson, "Identifying redundancy in source code using fingerprints", Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative.

[6] B.S. Baker, "On finding duplication and near duplication in large software systems", Proceedings of the 2nd Working Conference on Reverse Engineering.

[7] T. Kamiya, S. Kusumoto, K. Inoue, "CCFinder: Multilinguistic Token-Based Code Clone Detection System for Large Scale Source Code", IEEETransactions on Software Engineering.

[8] Kodhai.E, Perumal.A, Kanmani.S, "clone detection using textual and metrics analysis to figure out all types of clones "in ITC, 2010.

[9] R. Sivakumar, Kodhai.E "code clone detection in website using hybrid approach", in IJCA(0975-888) volume 48-No.13,june 2012.

[10] S .K.Abd-El-Hafiz "A Metrix Based Data Mining Approach For Software Clone Detection", proc. IEEE 36th international conference on computer software and

application,2012.

[11] A.Kaur, B.Singh, "Study on Metrix Based App00roach for Detecting pppplppppSoftware Code Clones" ISSN:2277 128X 2014.

[12] R.Koschke, R.Falke, Pierre Frenzel "Clone Detection using Abstract Syntax Suffix Trees"-working conference on Reverse Engineering-2006.

[13] P.Bulychev and M.Minea, "Duplicate Code Detection using Anti-Unification", A Survey on Software Clone Detection Research, 2007.

[14] J.krinke, "Identifying Similar Code with Program Dependency Graph", proc.eighth working Conf. Reverse Eng, 2001, pp 301-309.

[15] R.komondoor and S.Horwitz "Using Slicing to Identify Duplication in Source Code", proc. Int. Symp. Static Analysis, 2001.

[16] Rowyda Mohammad AbdEl-Aziz,Amal Elsayed Aboutabl,Mostafa-Sami Mostafa , "Clone Detection Using DIFF Algorithm For Aspect Mining" , International Journal of Advanced Computer Science and Applications, Vol. 3, No.8, 2012

[17] Saif Ur Rehman,Kamran Khan,Simon Fong,Robert Biuk-Aghai , "Hidden Cluster Detection and Visual Data Mining Framework for Infectious Disease Control andQuarantine Management – II", joint project with Dr. Si Yain Whar (FST), January 2010 – December 2011

[18]Praveen Ranjan Srivastava, Sirish Kumar, A.P. Singh, G. Raghurama." Software testing Effort: An Accessment Through Fuzzy Criteria Approch" Journal of Uncertain Systems , 5, 183-203, 2011

[19] Sahil Batra, Dr. Rahul Rishi "IMPROVING QUALITY USING TESTING STRATEGIES". Journal of Global Research in Computer Science , 2011.

[20] Khan, M. E. " Different Forms of Software Testing Techniques for Finding Errors". IJCSI International Journal of Computer Science Issues , 7, 2010.

[21] Samuel A . Ajila , Angad S,Gakhar, Chung H.Lung, Marzia Zaman , "Reusing and Converting Code Clones to Aspects - An Algorithmic Approach", IEEE IRI 2012

[22] Dr.gayathri Devi , Dr.M punithavalli **, "** Developing a Novel and Effective Clone Detection Using Data Mining Technique" , International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 8, August 2012 ISSN: 2277 128X.

[23] Salwa K.Abd-El-Hafiz , "Code Cloning: The Analysis, Detection and Removal" , International Journal of Computer Applications (0975 – 8887) Volume 20– No.7, April 2013.

II. Websites

[24] http://guru99.199tech.com/software-testing-introduction-importance.html

[25] http://www.slideshare.net/engineerrd/software-requirement

[26] http://arxiv.org/abs/1205.5615