LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

Improving the quality of software cohesion metrics through dynamic analysis

A Dissertation Proposal

Submitted

By

Ravneet Kaur

11304222

To

Department of Computer Science & Engineering

In Partial Fulfillment of the Requirement for the

Award of the Degree of

Master of Technology in Computer Science Engineering

Under the Guidance of

Ms. Rupinder Kaur

Asst. Professor

(MAY 2015)

ABSTRACT

Class cohesion is an important object oriented software quality attribute. Assessing the class cohesion and improving the class quality accordingly during the object oriented design phase allows for cheaper management of the later phases. Most of the cohesion metrics planned in the journalism describes static cohesion measurement at design level. Static metrics is a measure that is applied at the design level. Dynamic metrics is some measures that are performed at object level. It provides more accurate cohesion value as compared to the static metric. Dynamic cohesion performed measurement and provide good approach into behavioral aspects of the system. Dynamic cohesion measurement provides the capacity of cohesion metric at object level and using various object oriented characteristics such that encapsulation, polymorphism, parameters for the period of measure. Dynamic cohesion metrics introduce the correct meaning of the measurement and then define measures, validation and verification. The dynamic analysis construct examine tool and performing dynamic analysis of java programs for collecting the dynamic data for evaluation and measures the dynamic cohesion. The dynamic metric defined accurate distinction in between the existing metric into previous one. The dynamic cohesion metrics are validated by using byte code, java applications to find the affected class and performs changes into it. The execution based approach is used to measure module cohesion of legacy software. The cohesion metric is based on definition-use pairs in the dynamic slices of outputs. This approach significantly improves the accuracy of cohesion measurement. The dynamic technique is applied on the static metrics and measures module cohesion that will give more appropriate values of cohesion.

CERTIFICATE

This is to certify that Ravneet Kaur has completed M Tech Dissertation proposal titled "**Improving The Quality Of Software Cohesion Metrics Through Dynamic Analysis"** under my guidance and supervision. To the best of my knowledge, the present work is the result of her 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 conditions for the award of M Tech Computer Science & Engg.

Date: Signature of Advisor

Name:Rupinder kaur

UID:D6-16835

DECLARATION

I hereby declare that the dissertation entitled "**Improving The Quality Of Software Cohesion Metrics Through Dynamic Analysis**" submission 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 : Investigator :

Ravneet Kaur

Reg. No. 11304222

ACKNOWLEDGEMENT

First of all, I would like to thank my Almighty God, who has always blessed me and for giving me strength to do this work.

I wish to express my deep gratitude to my guide, Ms. Rupinder Kaur, for her generous guidance. Her guidance and support throughout all stages of the thesis process enabled me to conduct the research. Without her support, I would not be possible to complete this program.

Then I would like to offer my sincerest gratitude to Mr. Dalwinder Singh, Head of the Department, Computer Science & Engineering, for providing all the facilities and environment.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

Software engineering is the application of progress, advancement, maintenance and implementation of software. Software engineering is split into ten various parts as follows:

- Requirements engineering: The investigation, description, verification of needs for software components.
- Software design: It is defined as the procedure of construction, mechanism, interfaces, and some characteristics of a software system or component.
- Software construction: This is complete building of meaningful software by combining coding, validation, testing.
- Software testing: The object level certification is a program of set of test cases is satisfied to select unlimited domain execution and contrasting to the expected performance.
- Software configuration management: In configuration management is defined as systematically controlling the configuration variations and maintaining the integrity in software life cycle.
- Software engineering management: The requisitions of organization are-arrangement, combination, and reported-to certify that the software construction and implementation of the software is systematic.
- Software engineering process: It is defined as the major key parts of the projects such as implementation, maintenance, estimation, development and quality.
- Software engineering tools and methods: This is one tool that is related to the software engineering tool. It provides the awareness about the life cycle. To make the software engineering activities in systematically way.
- Software quality management: It is the ratio of inherited properties that fulfills the customer needs.

In recent times, software projects have faced many risks in using poor quality metrics during the evaluation of project performance. Mostly developed by errors, failures, variations, backtracking to the requirements of various issues like cheap, bad demand, not fulfill the requirements and pointless requirements are reached. Requirement management is better practiced to increase the demand of acceptance, reduce overhead and also improve the success rate. Projects measurement intended at quantification and evaluated presentation of a project and could be employed to evaluate the product error and assume the quality. Basically focus on the measurement which is additional necessity of software management. To add some additional supplement measures for requisite performing better estimation of software project. As re-engineering is in demands on commerce functions and information technology that supports them are changing at a rapid rate in the commercial organization. Many organizational recovering the corporate memory, domain knowledge, rules etc due to effort of re-engineering. It strives to break away from old rules and organized the changes in their old rules according to new techniques. Companies have conducted their maintenance according to new technology and techniques. Due to increased competition in technology, we adopt innovative approaches to renovate their legacy system with respect to process, product and service because the cost to maintain new software is too much high and a more time is needed. According to requirements to take economic values to reduce the maintenance cost. Cost effective software engineering requires identifying and measuring impact on system and different managerial tasks. This study includes the metrics of software engineering in different terms that are related to different tasks which are performed by software process model.

- To measure the quality and also measure correlation between specification and final product.
- To improve the quality of product by measuring the range of quality and cost of the software projects.
- To evaluate different methods and tools which are used to measure the performance, productivity, reliability and quality.
- To improve the reliability of software projects by measure the mean time to failure.

These strategies are applied as project planning begins. Software function, provides the evaluation detail from ending to starting of estimated for quality, cost and scheduling which is function oriented. For examples, Due to decomposition, we have to start from top go through at bottom end to analysis the whole moments. The advancement in different languages and software technology helps the programmer to estimate time and cost, quality efforts for software development. A processes triangle obtainable within a round of environment situation that included the metrics atmosphere as related to CASE TOOLS and techniques and customer characteristics. The metrics of software engineering should follow the business rules and does not cross the deadlines of software project. The customer satisfaction should follow the quality of software project and the maintenance cost of software should be affordable range. The goal of software project management is to understand the current software within its specifications, design and implementation, and then to re-implemented it to improving the systems function and performance. The functionality of existing software project also prepare for the functionality to be added later, this functionality is used for further enhancement in organizational approach.

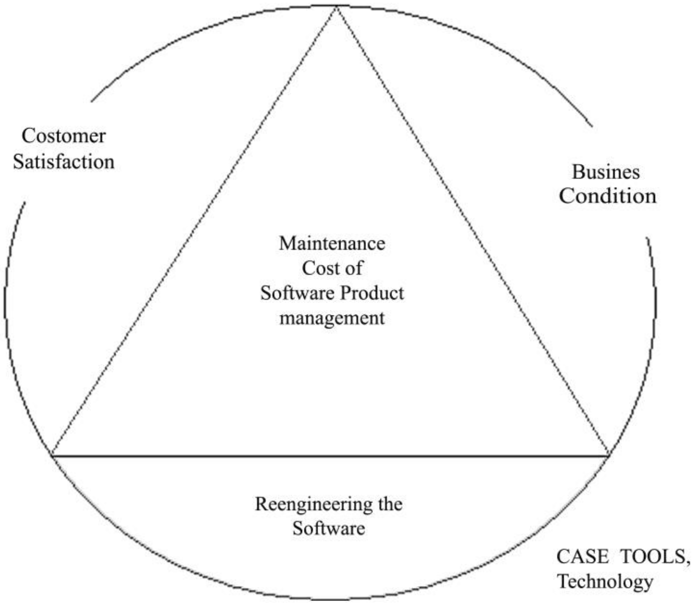


Fig 1.1CASE Tools Technology

The software project management programmed for software system. The software projects program establishes goal-driven steps:

- Identify the software system which is used in our organization.
- Identify the software system to which tools and techniques are used in further Maintain the metrics or software engineering models.

- Identify the entities and attributes related to our software project model.
- Formalize the measurement goals.
- Prepare a plan to which requirement analysis techniques and technologies, business goals are applied for the maintainability.
- The system should be maintained within time and cost and quality of project management model.
- Identify the data elements and construct the indicators which will help in future for maintain the software project.
- High priority changes are implemented and support for other software product in an easy manner.
- The maintained software project model should access the quantitative characteristics of cost, size and time to develop the software project.

1.1 Definition of Metrics:

It is very important views of organization according to consultant Peter Drucker. If you Cannot measure, if the developer and manager are not able to measure precisely, it means they could not evaluate the performance of the software projects. The success rate have been best achievement of the companies, yet the measures that provide benefit of measurement the success or project failures are very dissimilar and these type of metrics scarcely has a good cohesion. Utilizing congruous quality metrics is critical to easily manage the projects, as on the other hand it can be hard for a project manager to find out that project is improving according to the adaptation. Measurements are involved current situation of the projects and evaluate its strength. Measures are evaluated project conditions and divided into requisites, hazards, errors, testing and documents. When metric is not defined at a task level rather it is a combination of various metrics that can represents the behavior of the project.

1.2 Metrics to Manage Efforts:

An attempt is defines as: a strain of advantages or authority, either a material or rational in performed less number of actions on the objects and to achieve the relevant results. In generic term, total numbers of time is needed the effort to achieve the results during the development of product. To carried out the total number of times to achieve a results called plan effort. And estimated the total times that can spend to achieve an actual result. Effort or force calculated in term of times like minutes, hours, seconds and days.

1.3 Quality Metrics:

Basically Quality is the ratio to which condition is applied to describe the product design. Quality is finding the error in the whole products and fact is shows at the later stages. That is the reason, Quality is describe the cruel, low defects is delivered all the time in life time projects.

1.4 Productivity Metrics:

Productivity is basically a process define the competence number of inputs is applied and produced the number of outputs. Productivity is normally measurement by the total ratio of output to inputs; it means how many inputs are taken and producing the outputs. When the percentage of indication is increased then productivity is also increases, but indication is decreased then productivity is also deceases. Productivity is the collection of number of simple tasks.

So that, a best clarification of productivity could be describes the time is needed a particular variables to delivered the output in hours that is six hour. Then getting into account the six working per day in hours, the productivity could be evaluated are as follows,ratio:Productivity=((totalPlannedEffort/4)/ActualEfforts)6 ………..eq.1.1

Size Metrics: The software size physically measures the length of software system's code and design. Size metrics has following attributes:

1. Vocabulary Size: It counts the number of components. It helps to calculate number of classes and its aspects. This metric is deal with the system vocabulary size. Each component name is counted as part of the system vocabulary. The component instances are not counted.

2. Line of Code: It counts the number of code per line. It is used to measure the size of the code. It was the traditional method for cost estimation.

3. Number of Attributes: This helps to count the number of attributes of each class and aspects. It plays important role to calculate internal vocabulary.

4. Weighted operation per components: This metric measures the complexity of a component in terms of its operations. Consider a component C1 with operations O1 ,..., On. Let c1 ,..., c_n be the complexity of the operations. Then:

$$WOC = c1 + ... + c_n \qquad …….eq.1.2$$

This metric originally does not specify the operation complexity measure, which should be tailored to the specific contexts. The operation complexity measure is obtained by counting the number of parameters of the operation, assuming that an operation with more parameters than another is likely to be more complex. This metric extends the CK's WMC metric. It is an advice and methods of aspects in the same way that CK treats methods of classes.

1.5 Effective Measurement

Asthana and Olivieri described the shows various four key mechanism of a capable measuring procedure.

- Defined life cycle issues obviously and the software products measurement that hold approaching to issues.
- Flow of data in graphs form and tabular forms.
- Data is providing to analyzing and approaches to many issues.
- To improving the implementation of output results and to produces various approaches.

1.6 Cohesion: cohesion metric measures how many the methods and class functions are inter-related to each other. A cohesive class performing not more than single function, it means using one function. A non-cohesive class performed more than two non-related functions. A non-related function class might required be constructed again multiple similar classes.

There are several types of cohesion:-

Coincidental Cohesion: It implies having no any relation within the statement code and procedure. They interact with the co-incidents.

Logical Cohesion: All elements of the modules are related to each other with logically. All the elements of the module components are similar to each other.

Temporal Cohesion: Al the elements of the components are related with timing.

Example: Shut down closes all the files.

Procedural Cohesion: Procedural cohesion is performing functions not in sequence, but given to by step by step process is carried out the objectives.

Example: algorithms.

Communication Cohesion: In communication cohesion defines all the files of the elements have data structures. Example: array, stack.

Sequential Cohesion: The elements of the components perform sequential operation; it means each part of the elements output is the input of the next stage.

Example:

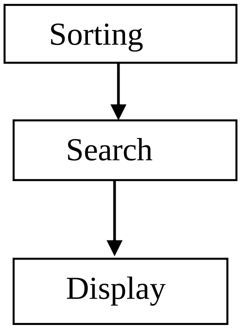


Fig 1.2 Example of sequential cohesion

Functional Cohesion: There are different components of the modules to share a same functions and cooperating with each other.

Basic idea of related class: A related cohesive class performed only a single function. If non-related function class performed un-related functions, it would be split into them.

- High cohesion is preferred it assist encapsulation. Drawback, a related function class includes highly coupling in between the modules method.
- Low cohesion indicates poor design.

In case of dynamic cohesion metrics, basically is performing a dimension. Dynamic or object level cohesion metrics to measure the run-time level and provide the same class objects and all the objects lies at same class . The design based cohesion measured at class level. The capacity of run-time cohesion dimensions is always the whole part of class.

STATIC METRIC:

It is a measure that is applied at the design level. It measures the quantity and complexity of the cohesion at design time. Static metrics are less accurate than dynamic metrics.

Static metrics are provided only at early stages of the development. Static Cohesion metrics are:

RCI (Ratio of Interaction)

CAMC (Cohesion among Method in a Class)

NHD (Normalized Hamming Distance)

RCI (Ratio of Interaction) is a design level metric defines the data-data and data-subroutine interactions within the variables. RCI metric take the direct interaction into the accounts. CAMC (Cohesion among Methods in a class) it defines the method-method interaction within the variables. CAMC is following the DAT matrix having no. of ith rows and no of jth column. No. of rows define the no. of attributes and no. of column is defined the no of methods.

$$CAMC(C)\Omega/kl where\, \sigma = \sum\sum^{k}{}_{l} PO[i]^{l} \perp [j] \qquad \text{……..eq1.3}$$

NHD (Normalized Hamming Distance) it considers the method-method interaction within the variables. The pair of methods is considered the no. of two same parameters is equal.

$$\text{NHD} = 2/lk(k-1)\sum_{j-1}^{k-1}\sum_{i=j+1}^{k} c_{ij}$$

$$=1-2/lk(k-1)\sum^{l}{}_{j=1} X_j(k-x) \qquad \text{…………eq1.4}$$

D_3C_2 (The Distance Design-based Direct Class Cohesion) is a static metric is consider the method-method, attribute-attribute, attribute-method interactions within the variables. It is better metric as compared to CAMC, DCC, NHD, because D_3C_2 is taken better accurate and better consideration. D_3C_2 is providing the three interactions to calculate the cohesion values. Various interactions are:

MMAC (method-method attributes Cohesion) where k= no of methods, l= no of distinct attribute type,

$$\text{MMAC}=\sum_{i=1}^{l} x_i(x_i-1)/lk(k-1) \qquad \text{……………..eq.1.5}$$

$$0= \text{If } k=0 \text{ or } l=0$$

$$1= \text{if } k=1$$

AAC (Attribute-Attribute Cohesion)

$$\text{AAC} = \sum_{i=1}^{k} y_i\,(y_i-1)/kl(l-1)$$

$0=$ If $k=0$ or $l=0$, $1=$ if $l=1$ ……….…eq1.6

AMC (Attribute-Attribute Cohesion) either $k=0$ or $l=0$,

$\sum_{i=1}^{k}\sum_{j=1}^{l} m_{ij}/kl$ …………….eq1.7

D_3C_2 (The Distance Design-based Direct Class Cohesion)

$D_3C_2=K(k-1)MMAC(c)+l(l-1)AAC(c)+2lkAMC(c)/k(k-1)+l(l-1)+2lk$ ………….eq1.8

Dynamic Metric:

It is some measures that are performed at object level. It is much accurate as compared to the static metric. Dynamic metrics are only considering the dynamic behavior of the software products. Dynamic metrics are available at later stages of the development. Dynamic metrics also involves the object oriented aspects and its codes. Static metric is less specific in real as compare to dynamic metrics. Dynamic metrics is collected data direct at run time and evaluate its quality and its attributes.

Dynamic metrics are as follows:

SFC (Strong Functional Cohesion)

WFC (Weak functional Cohesion)

SFC is a module based Cohesion it consider the same pair for all the output of slicing.

WFC is also module cohesion it consider the same pair for more than one output slicing.

In SFC and WFC is a dynamic metric, using dynamic slicing approach on it.

Dynamic Cohesion Metrics: It shows the measurement for dynamic cohesion metrics using program execution approach based upon dynamic slices. They use dynamic slices of outputs to measure module cohesion. Module cohesion metrics based on static slicing approach has some issues in cohesion measurement. This approach limelight the limitations of static cohesion and introduce a new scheme for dynamic cohesion. It is of two types:

1. Strong Functional Cohesion
2. Weak Functional Cohesion

SFC is a module cohesion which is obtained from common diffuse pairs of each type common for all the output of slices.WFC is a module cohesion which is obtained from diffuse pairs of each type common for one and two output of slices

CHAPTER 2
REVIEW OF LITERATURE

In the previous time, the efficient using software projects and strategy have improving the range of software projects at major level. In this view point, Software Project is develop as new regulations and enclosed the new range thoughts across the methodology to managed very basic knowledge of the projects. The number of production measures or various methods is using to preserve the architecture, progress and exploitation. The production metrics are available at every time for software projects to fulfill the needs and supports.

The dynamic cohesion metrics, cohesion performing the measurement at execution level and cohesion metric of equal class is obtain by total cohesion values of all objects lies to one class. The design level cohesion measured at class level. The extent of dynamic cohesion to measures always the combine complete class. In opposite side, range of objective cohesion measure is even be specified to a single object belongs to a group at dynamic level.

[1] Shweta Sharma & Dr.S.Srinivasan (2013)[12] In this paper focus on the cohesion and coupling metrics. Software metric are generally used to evaluate the complexity of the software products. Firstly, in this paper introduce the cohesion and its types. Cohesion is measures the functional strength of the modules. Various cohesion types are coincidental cohesion, temporal cohesion, logical cohesion, procedural cohesion, communicational cohesion, sequential cohesion, functional cohesion. Various coupling measures are described in this paper are Chidamber and Kemmerer suite of metrics is validate the software metrics. It validates the aspects of the object oriented technique, to validate its complexity. CBO (Coupling between Objects), RFC etc are the static coupling metrics. Several Cohesion measures are Import coupling metrics and Export cohesion measures. WFC and SFC cohesion measures using program execution technique definition use pair (slicing based programming technique) to calculate the cohesion values.

[2] According to **Varun Gupta, jitender kumar Chhabra (2011) [9]** paper, Dynamic cohesion metric discussed the metrics at execution level take into significant and used object-oriented various features such as encapsulation, array binding during the measurement. This paper focused on the previous study and discussion about dynamic metrics provides the dynamic behavior. Again defined a component of a unit cohesion metrics are SFC (strong functional cohesion), WFC (weak functional cohesion). The expectation of the dynamic metrics are shows the run time performance on the dynamic slicing. They are used dynamic slices of outputs to measured unit of cohesion. According to author define SFC dynamic metric is module cohesion obtained from common definition-use pairs of each type common to the various dynamic slices of the output variables and WFC metric cohesion obtained from definition-use pairs of each type find in dynamic slices of multiple object values. Mitchell and Power [7] define dynamic cohesion based on the Chidamber and Kemere LCOM (Lack of cohesion metric) and used AOP (aspect oriented programming) approach.

[3] According to **N. Sasrirekha, A. Edwin Robert and Dr. M. Hemalatha (July 2011)[5]** paper, Program dynamic slicing is a system for expressing the parts of software programs by as well as same data items of control flow and data flow. This paper focused on the various slicing plans such that class slices quasi static slicing, object slicing and conditional slices. At some stage in program slicing, the slicing principle contains the object values which producing an unpredicted result in the form of outputs on some input values to the program. Dynamic slicing takes the input absolute to the program for the duration of execution or run time and the slice contains only the statement that causing the failure for the period of the specific execution of interest. Dynamic slicing used dynamic analysis to recognize all and only the statement that affected the variables of awareness on the particular asymmetrical execution hint. The benefit of dynamic slices is the run-time array handling and pointer variables .Dynamic slicing can be treated all element of an array independently, but static slicing considered every definition or use of any type of array element as a definition or used of the absolute part of group. Dynamic slicing is distinction between the objects that are narrowed to by pointer variables during a program object level. Dynamic slicing criterions specified the values, and differentiate between different occurrences of a report in a run time.

[4] According to **Jitender Kumar Chhabra and Varun Gupta (2010) [10]** mainly this paper is presents the comparison of static and dynamic metrics. In this paper dynamic metric is better than the static metrics. In static metrics is less accurate than the dynamic metrics in terms of quantity, complexity and results. Basically static metric perform some measures as the design level and dynamic metric check the dynamic behaviour of the software at run time. In this paper also focus on the dynamic coupling metric and dynamic cohesion metrics. It defines the dynamic coupling metrics like EOC (Export Object Coupling), IOC (Import Object Coupling), CBO (Coupling between the Objects), and DCM (Dynamic Coupling Metric). Dynamic Cohesion Metrics is defined the WFC (Weak Functional Cohesion) and SFC (Strong Functional Cohesion) begin the dynamic cohesion metric using dynamic slicing approach to calculate the cohesion values. Static metric is accessible at the early stages of the development. Static metric is less accurate the dynamic metric. . Dynamic metrics are only considering the dynamic behavior of the software products. Dynamic metrics are available at later stages of the development. Dynamic metrics also involves the object oriented aspects and its codes. Static metric is less specific in real as compare to dynamic metrics. Dynamic metrics is collected data direct at run time and evaluate its quality and its attributes.

[5] According to **Dr.Linda H.Rosenbar (2010)[15]** this paper, Software Requirement Engineering and Process Model, due to use of COTS package is seen as a way to increase reliability with decreasing development and test time. Translation of code is means of decreasing time and cost. Metrics are also needed to re-engineering process, could be measured and that the metrics would infect evaluate what they are proposed to quantify. The resulted combination of development methods for cost estimation models hybrid the results of different parameters used to maintain the cost estimation. The quality of system will impact over the cost of re-engineering maintenance, because quality of software describes the technical domain risk engineering framework to which is applied to a categories functional and quality risk components of technical domain and to measure cumulative effect of different components. These different components of software relate to maintenance cost re-engineering, because within time period, the quality of different components must be maintained because latest technology have improved the overall impact of software project but different frames, domains of quality perspective involve the different maintenance models which the quality of software system within time.

[6] According to **Payal Khurana & Puneet Jai Kaur (2009)[3]** this paper, cohesion metrics measures only pattern Interactions but does not shows the variation any write interaction from read interaction. Thus, does not reflect properties of the class. This research measures the improving the cohesion measurements considering read and write interaction in addition to dynamic environment. In this paper, defines the interaction in between the multiple variables. LCOM1 and LCOM2 is the dynamic metric based on the object oriented programming approach.LCOM1 counts the digit of non-related two methods, so LCOM1 is computed by subtracted the number of similar type pairs of methods from the complete amount of distinct method pairs.LCOM2, amount of related two methods are subtracted from the amount of non-related method pairs. TCC (tightly class cohesion) considered the two different methods are inter-relate if the share some common occurrence variable in using. The previous version of TCC (tightly class cohesion) is considered the degree of the total addition of coherency heaviness of each group of methods to the relative amount of method pair off. As an expectations work, this work can be extending for the previous cohesion measurement.

[7] According to **Jehad Al Dallal (2007) [2]** this paper, Cohesion class is an important object-oriented quality defined quality attribute values. It shows how many participant of one class are similar to that group of class. Assess the class cohesion and getting improves the class feature considered the object-oriented during the static level. The metrics are considering the method-method, attribute-attribute, and attribute-method interactions. Attribute- attribute and attribute-method direct interactions allowed for cheapest management of the previous phases. According to this paper defined classes of attribute and method in various classes such as one method-to another method, one attribute-to another attribute values and one attribute connect to the method. The introduced metrics can be improving in several instructions, like considering not a direct interactions and method invocation interactions. A cohesion method-method interaction is representing in the DAT by two inter-related rows share binary values 1 in a column. Similarly, a cohesion class attribute-attribute interaction is representing in the DAT by two. The metric used the distance in between the pairs of methods and pairs of attributes as based to calculate their degree of similarity. According to author several predefined design-based class cohesion metrics are overviewed and discussed.

Design-based class cohesion Metrics are: - Ratio of cohesive Interactions (RCI).This metric includes the data to data and data to subroutine interactions. Briand et al. consider each definition of a quality variable of a kind defined within the software components a cohesion interaction between the variable and the category. Interactions in between the variables within subroutines are not included, since detailed parts are not included at static level. The data to subroutine interaction is occurring, if a type define within the software component matching the kind of one subroutine parameters values, or an quality variable within the software part is scheduled in the method constraint list. The RCI metric is defined as the ratio of the number of cohesive interactions of a module to the proportion of possible cohesion interactions. The RCI metric do not get the indirect interactions into account. In addition, Briand et al. considered the insertion of method invocation interactions and inheritance relation as subjects for potential work. Cohesion among Methods static class cohesion metric called Cohesion among Methods in a Class (CAMC).In this metric, only the method-method interactions are including. The CAMC metric uses a parameter incidence metrics to have a row for each method and a domain for each facts type that appeared at smallest amount one time as the style of a parameter in at least only single method in the class. The CAMC metric is defined as the sum number of 1s in the metric to the total dimension of the metric.

Normalized Hamming Distance (NHD) Metric- static class cohesion metric called the Normalized Hamming Distance (NHD). In this metric, simply the one method to other method interactions is including. The metric uses the same single constraint incidence metric used by CAMC metric. The metric is calculating the average of the parameter agreement within the every pair of methods. The constraint agreement between couples of methods is defined as the number of places in which the parameter occurrence vectors of the two different methods are same.

The D_3C_2 (The Distance Design-Based Direct Class Cohesion) metric is a static metric, In this Paper D_3C_2 static metric is much reliable metric as compare to CAMC, NHD, RCI metric, D_3C_2 metric having more sensitivity and more commonality in cohesion values. It provides more accurate results.

Table 2.1.Comparison of static metrics

Class	CAMC	NHD	D_3C_2
1	0.2	0.6	0.16
2	0.2	0.6	0.11
3	0.36	0.68	0.29
4	0.36	0.48	0.26

[8] According to **Fernando Brit et.al (2004) [16]** this paper, the object-oriented paradigm brought a new design philosophy and encapsulation mechanisms that apparently would help us to achieve that desideratum. However, after a decade where this paradigm has emerged as the dominant one, we are faced with practitioners' reality: coupling and cohesion do not seem to be the dominant driving forces when it comes to modularization. This conclusion was based on a relatively large sample of heterogeneous systems. They describe an environment that allows not only assessing this reality but also deriving better modularization solutions in what concerns coupling and cohesion. These solutions are generated by means of cluster analysis techniques and partially preserve the original modularization criteria. They believe this approach can be of great helping reengineering actions of object-oriented legacy systems.

[9] According to **Neelam Gupta, Parveen Rao (2001) [6]** paper, the high module cohesion is a popular property of a software program. In this paper, the software program object based approach to quantify the module cohesion of heritage software. We define cohesion metrics based on definition-use pairs by using dynamic slices of the outputs. This approac is significantly improving the correctness of cohesion measurement. Def-use pair is dynamic technique used for dynamic analysis.The use of definition-use *pairs* allow us to measures the functional cohesion is more precisely by include all attribute variables references that provide towards functional cohesion. Program execution based approach uses the definition-use pairs on dynamic slices of outputs to measurement of the functional cohesion. Cohesion measurements are based on the def-use pairs on dynamic slices of outputs; we executed the function, firstly collection of data and organised it. Then measure the cohesion metrics. The first set of metrics measured the cohesion in between the output variables of the function. We define Strong Functional Cohesion (SFC) as that arising out of def-use pairs of every type common to the dynamic slices of all the outputs variables.

[10] According to **Claudio Nogueria Sant Anna et al (2009) [11]** this paper is discuss the aspect-oriented software development (AOSD) approach. Aspect-oriented approach is taking better attention in research cloud and in industrial. In Aspect-oriented approach considers the data abstraction and any other complexity proportions. In software engineering aspect oriented is needed because it measures the reusability, maintainability, reliability and complexity. Basically AOSD is defined the components that are metric suits and model of quality. These all components are based on recognized principle that is avoiding delicacy in test solution. In proposed research framework has been calculated in the background experimental study with different function/in this paper has been discussing the experimental study and its analysis, and also is drawbacks and benefits are discussed under it.

[11] According to **Marcus, A. (2005) [12]** paper cohesion is used in software for the measurement work. Software cohesion is using measures to quality and proneness of the software modules. Various approaches to cohesion measure the interactions in between the variables. In this paper is proposed many other measures for the measures the cohesion of single class and organized into modules. In paper case study is compare the existing measures within the new ones. The drawbacks and advantages among the object oriented design approach and results are discussed.

[12] According to **Challa Bonja(2006) [13]** this Paper, is focused on the static cohesion metric and object oriented designs. Managers and developers need to a good metric to compare analyse and evaluate the solution to solve a problem. Class cohesion metric is based on the similarities between the methods. In research, metrics are develops as the bases on the class similarities in the methods. Class cohesion metric is developed to evaluate the set of characteristics of measurements. Class cohesion metric is introduced the LCOM, CAMC metrics. These are the more powerful metrics is used for the measurements. These metrics are defined the cohesive class and non cohesive class.

CHAPTER 3
PRESENT WORK

3.1 Problem Formulation

The software complexity can be estimated with the help of cohesion and coupling values. The cohesion is degree to which the different component of a class interacts with each other. The importance of the cohesion value can be judged at the time when the software is maintained. When some software is developed and installed on the user end at that time the cohesion value will be calculated by the developer and it desired to be high. For the software maintained purpose you need to change the software modules due which the cohesion values are reduced. In the previous time various static techniques had been applied to calculate cohesion value. The static technique will not able to exactly calculate the cohesion value which will reduce the efficiency of cohesion metrics. The definition use pair technique is the slicing technique which is applied to calculate cohesion value, but the static slicing will either overestimate or underestimate cohesion values. Dynamic metric is better than the static metric, because it provide the accurate calculation of the cohesion values. In this work, we apply dynamic technique to calculate dynamic cohesion exactly when some changes will be applied in the software. In this research, are using some dynamic technique that is applied on the some static metrics. Dynamic techniques are ontology of object include interaction pattern and Def-use pair includes dynamic slicing method to calculate the measurement of the metrics. In present research, is using definition use pair dynamic approach on the static metric D_3C_2 (The Distance Design-based Direct Class Cohesion).

3.2 Objectives

In past, SFC (Strong Functional Cohesion) and WFC (Weak Functional Cohesion) is static metric used definition use pair dynamic approach has got some inadequacies in cohesion measurement. The static measures significantly over estimate the levels of cohesion present in the software. But in the present research uses D_3C_2 (The Distance Design-Based Direct Class Cohesion) on which dynamic technique is applied to achieve better and accurate results. Earlier, very few researches had been performed under it. The dynamic cohesion measures are more accurate as they are defined at run-time and take into proper consideration the factors that change at run time.

- To study various static cohesion metrics like CAMC, NHD, DCC and D_3C_2 (The Distance Design-based Direct Class Cohesion) etc.
- To calculate cohesion using static approach i.e. D_3C_2 metrics.
- To study various dynamic techniques for cohesion measurement like Def-use pairs (Dynamic slicing), ontology of objects etc.
- To perform dynamic analysis of D_3C_2 using a new metric that is D_4C_2 (The Dynamic Distance Design-based Direct Class Cohesion) for dynamic calculation of cohesion.
- To compare results of D_3C_2static metric with the new D_4C_2 metric.

3.3 Scope of the Study

The object oriented software is based on the objects through which we can easily analyze the quality and design of the software. In the previous times much research has been done to properly analyze the design and structure of the software. To analyze the design and structure of the software, coupling and cohesion values are used, but these values are calculated with the static metrics which reduce the efficiency of the software analysis. To properly analyze the software various dynamic techniques had been proposed in the previous years. To study various dynamic techniques are ontology of object and definition use-pair. Ontology of object includes the interaction pattern and definition use-pair includes dynamic slicing. In this technique, an enhancement in cohesion metrics is done, in cohesion metrics they considered the prototype of interaction but do not any distinctions write interaction from read interaction. In the enhancement they consider the write and read interactions dynamically and calculate cohesion values. In this research, comparison of D_3C_2 is done with the new metric D_4C_2. Dynamic metric is better than the static metric, because it provide the accurate calculation of the cohesion values. In this work, we apply dynamic technique to calculate dynamic cohesion exactly when some changes will be applied in the software. The static technique will not able to exactly calculate the cohesion value which will reduce the efficiency of cohesion metrics.

3.4 Research Methodology

In this work, calculation the cohesion value of the software modules is done dynamically using definition use pair approach. The definition use pair is technique which is based on the dynamic slicing. The static metric which we use is D_3C_2 to calculate cohesion values. Dynamic technique is definition-use pair that is used for the dynamic analysis. This technique defines dynamic slicing to measure the accuracy of the cohesion metric. Definition use-pair approach includes dynamic slicing of the java programs. Firstly, in step 1: To obtain set of unique dynamic slices (UDS) for each function outputs.

Step 2: To analyze information to identified commonality between definition use-pair in the dynamic slices.

Step 3: To compare the unique dynamic slices (USD) with the commonality.

The advantages of dynamic slicing technique are the array handling and pointer handling. Definition use pair technique is the dynamic technique can be applied at D_3C_2 (The Distance Design-based Direct Class Cohesion) to improve the software quality of cohesion metric.

- First of all study various static metric RCI, DCC, NHD, SNHD, CAMC, D_3C_2 and dynamic techniques Def-use pair.
- Choose D_3C_2 static metric to calculate the cohesion values, because D_3C_2 metric is better than the existing static metrics that is RCI, CAMC and NHD.
- Calculate cohesion values using definition use pair approach.
- To perform dynamic analysis of D_3C_2 a new metric that is D_4C_2 (The Dynamic Distance Design-based Direct Class Cohesion) for dynamic calculation of cohesion.
- To compare results of proposed D_3C_2 metric with a new D_4C_2 metric on the basis of obtained cohesion values.

  Firstly, to study various static and dynamic techniques to performs the dynamic analysis. To study various static metrics to calculate the cohesion values of a software. Various static metrics are RCI (Ratio of Cohesion Interaction), CAMC (Cohesion Among the Method of Class), NHD (Normalized Hamming Distance), SNHD (Scalar Normalized Hamming Distance), DCC (Direct Class Cohesion), D_3C_2 (The Distance Design-Based Direct Class Cohesion). D_3C_2 is better than the RCI, CAMC, DCC and NHD because in D_3C_2 metric show more sensitivity and commonality then the other metrics.

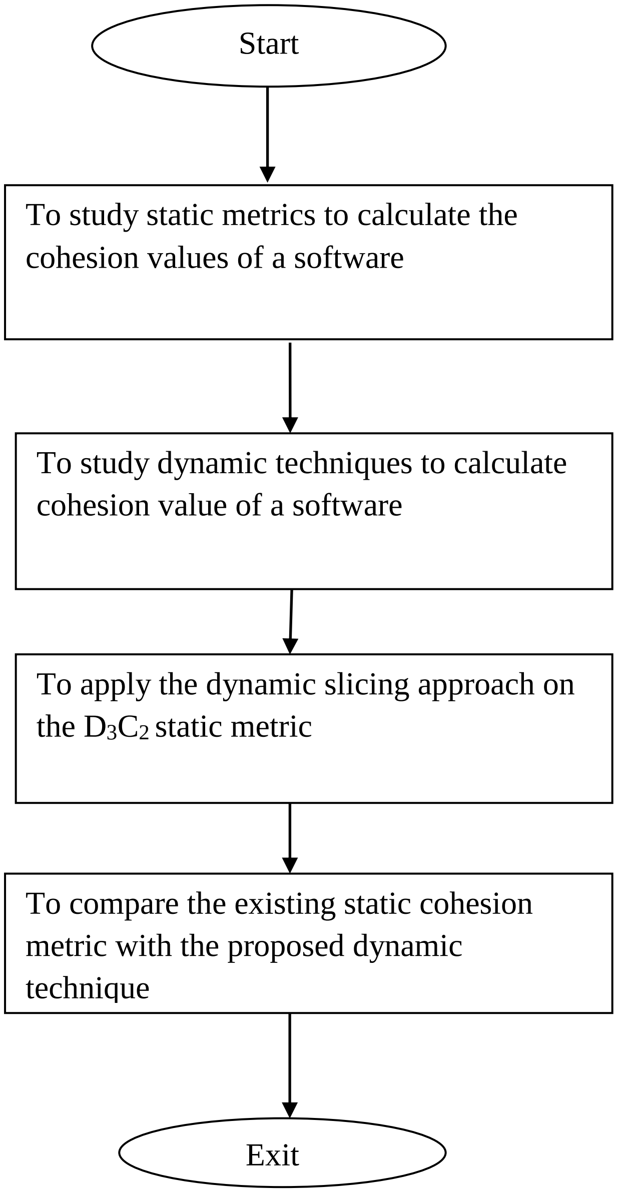

Start
To study static metrics to calculate the cohesion values of a software
To study dynamic techniques to calculate cohesion value of a software
To apply the dynamic slicing approach on the D_3C_2 static metric
To compare the existing static cohesion metric with the proposed dynamic technique
Exit

Fig 3.1 Flow chart of research methodology

In D_3C_2, take some java project are taken and then creating UML (Unified Modeling Language) for account dialog box.

Construct the DAT (Direct Attribute-Type) using the information is provided by UML diagram. In D_3C_2 metric is using three interactions to calculate the cohesion values. Three interactions are as follows:

MMAC (Method-Method through Attribute Cohesion)

AAC (Attribute-Attribute Cohesion)

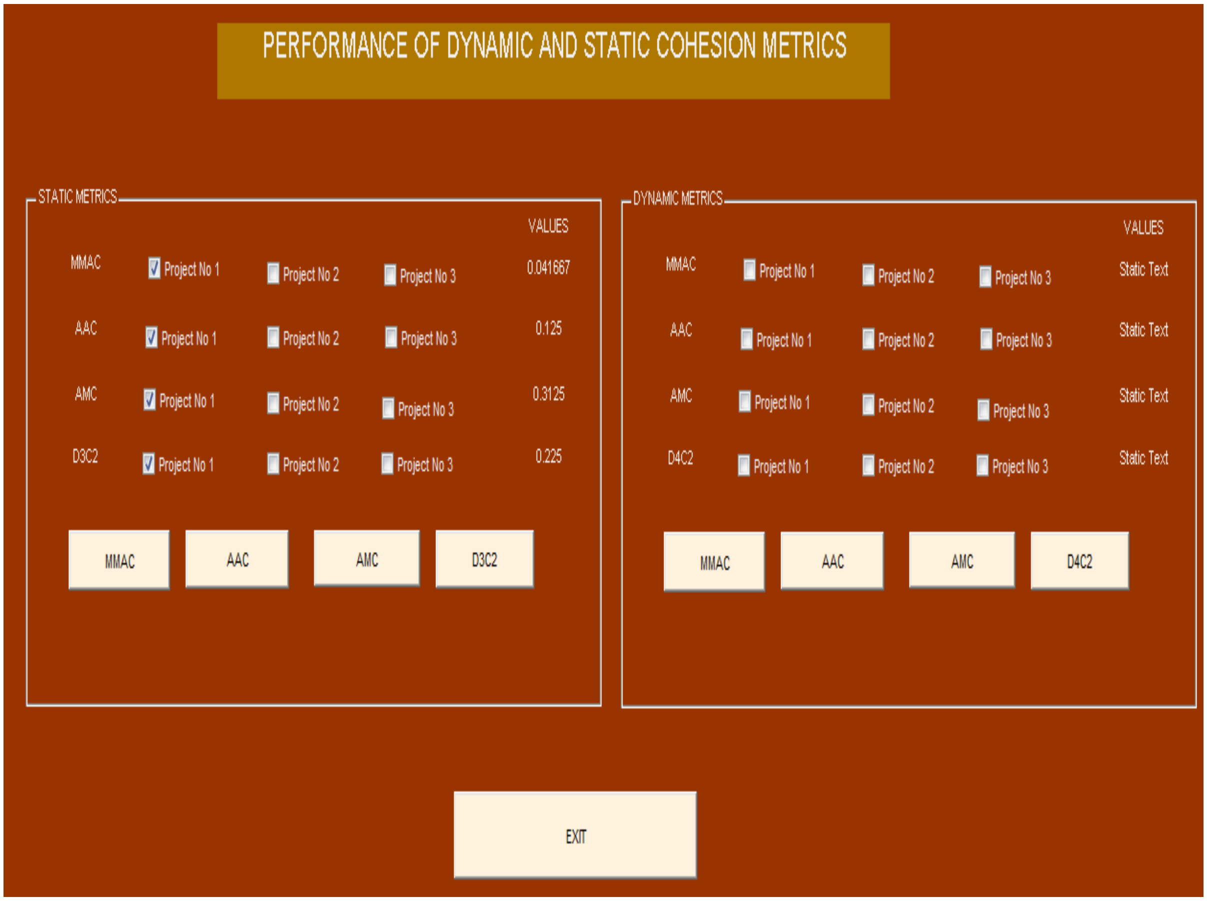
AMC (Attribute-Method Cohesion)

Calculate total cohesion: D_3C_2

Using dynamic approach to performs dynamic analysis of the existing static metric. Definition Use-Pair using dynamic slicing is calculating the cohesion values of static metric.

CHAPTER 4

RESULTS AND ANALYSIS



PERFORMANCE OF DYNAMIC AND STATIC COHESION METRICS

STATIC METRICS

				VALUES
MMAC	☑ Project No 1	□ Project No 2	□ Project No 3	0.041667
AAC	☑ Project No 1	□ Project No 2	□ Project No 3	0.125
AMC	☑ Project No 1	□ Project No 2	□ Project No 3	0.3125
D3C2	☑ Project No 1	□ Project No 2	□ Project No 3	0.225

DYNAMIC METRICS

				VALUES
MMAC	□ Project No 1	□ Project No 2	□ Project No 3	Static Text
AAC	□ Project No 1	□ Project No 2	□ Project No 3	Static Text
AMC	□ Project No 1	□ Project No 2	□ Project No 3	Static Text
D4C2	□ Project No 1	□ Project No 2	□ Project No 3	Static Text

Fig.4.1 Calculation of static cohesion for project 1

- ➢ D3C2 is the static metric used to calculate the cohesion value statically.
- ➢ In D3C2 having three calculation formula's to calculate the cohesion values, that are following:

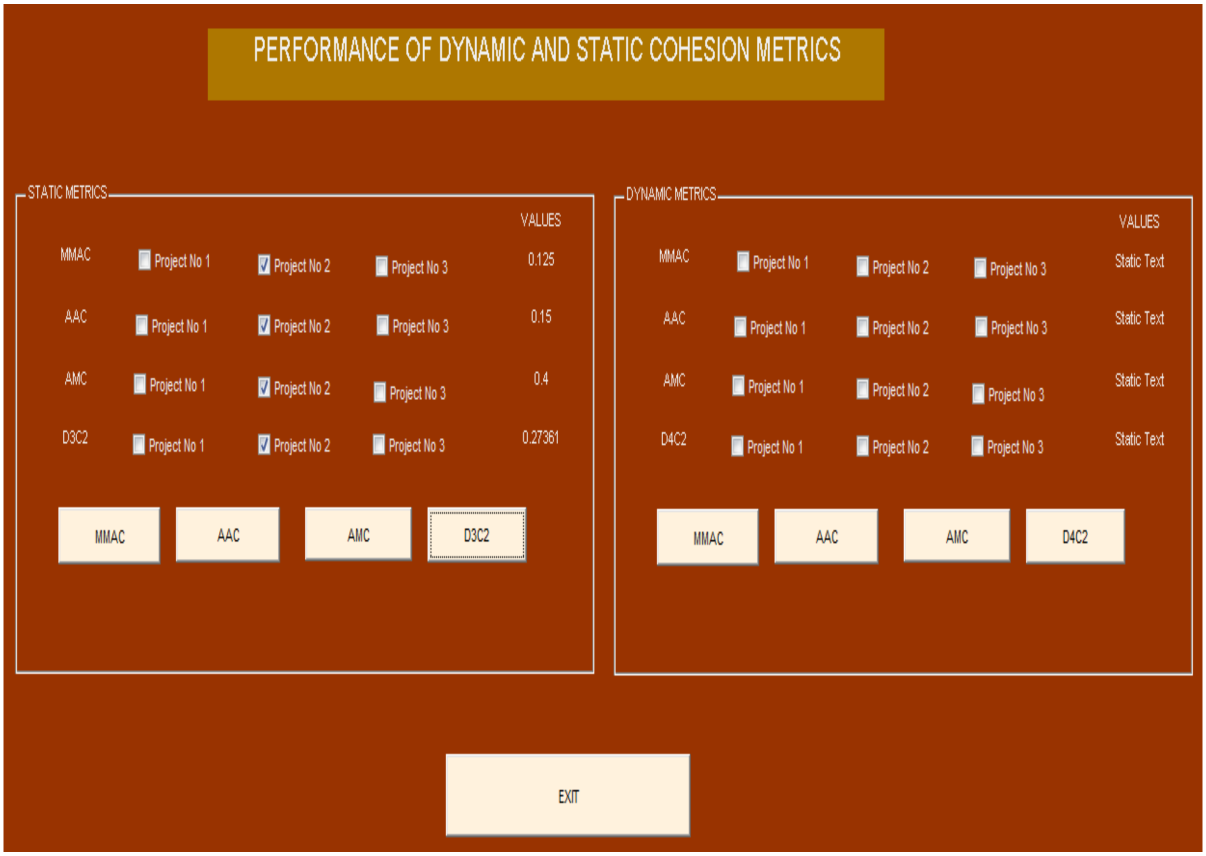
MMAC: Method- Method through Attributes Cohesion

AAC : Attribute-Attribute Cohesion

AMC : Attribute-Method Cohesion

D3C2 : The Distance Design-Based Direct Class Cohesion

- ➢ To select the project1 and to find the MMAC, AAC, AMC, D3C2.


PERFORMANCE OF DYNAMIC AND STATIC COHESION METRICS

STATIC METRICS

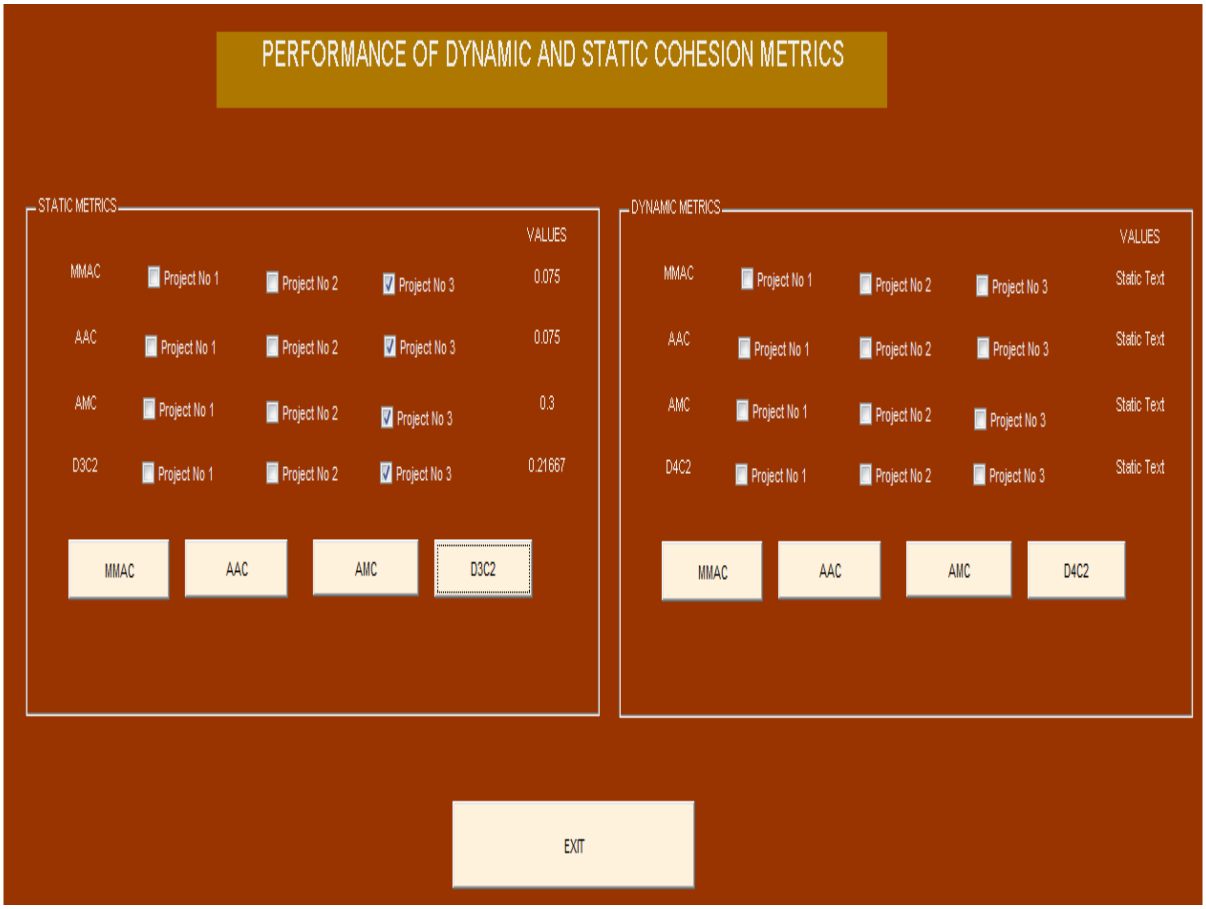
				VALUES
MMAC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.125
AAC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.15
AMC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.4
D3C2	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.27361

DYNAMIC METRICS

				VALUES
MMAC	☐ Project No 1	☐ Project No 2	☐ Project No 3	Static Text
AAC	☐ Project No 1	☐ Project No 2	☐ Project No 3	Static Text
AMC	☐ Project No 1	☐ Project No 2	☐ Project No 3	Static Text
D4C2	☐ Project No 1	☐ Project No 2	☐ Project No 3	Static Text

Fig. 4.2 Calculation of static cohesion for project 2

- D3C2 metric is used to calculate the cohesion value is statically for the project2.
- To select the project2 and calculate the MMAC, AAC, AMC and D3C2 values.
- To show the variation is in the values of the project1 and project2.
- MMAC, AAC, AMC, D3C2 checkbox is calculates the cohesion values on the clicking.



PERFORMANCE OF DYNAMIC AND STATIC COHESION METRICS

STATIC METRICS

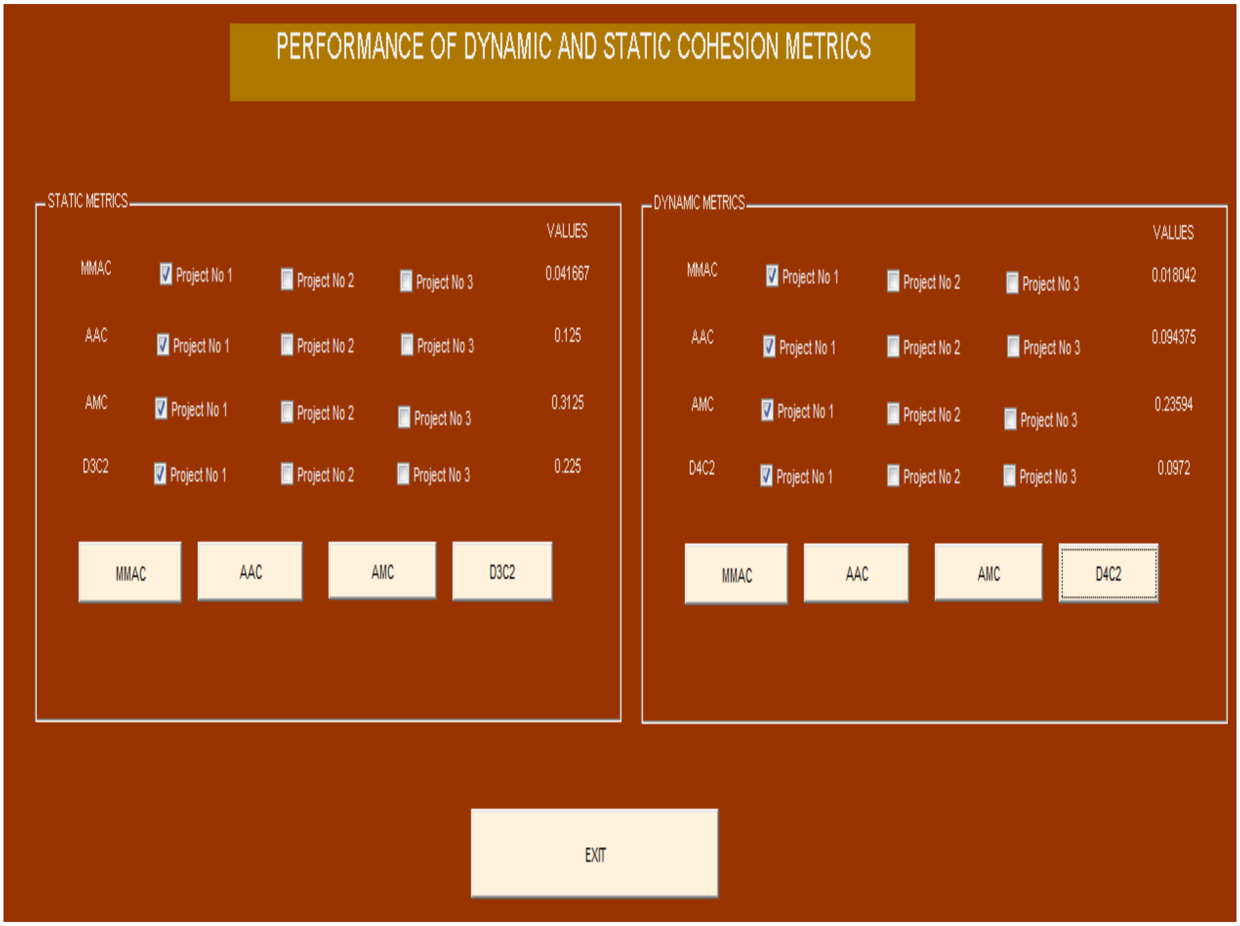
				VALUES
MMAC	□ Project No 1	□ Project No 2	☑ Project No 3	0.075
AAC	□ Project No 1	□ Project No 2	☑ Project No 3	0.075
AMC	□ Project No 1	□ Project No 2	☑ Project No 3	0.3
D3C2	□ Project No 1	□ Project No 2	☑ Project No 3	0.21667

DYNAMIC METRICS

				VALUES
MMAC	□ Project No 1	□ Project No 2	□ Project No 3	Static Text
AAC	□ Project No 1	□ Project No 2	□ Project No 3	Static Text
AMC	□ Project No 1	□ Project No 2	□ Project No 3	Static Text
D4C2	□ Project No 1	□ Project No 2	□ Project No 3	Static Text

Fig. 4.3 Calculation of static cohesion for project 3

- D3C2 metric is used to calculate the cohesion value is statically for the project3.
- To select the project3 and calculate the MMAC, AAC, AMC and D3C2 values.
- To show the variation is in the values of the project1 and project2 and project 3.
- Exit button is used to quit from this page.



PERFORMANCE OF DYNAMIC AND STATIC COHESION METRICS

STATIC METRICS

				VALUES
MMAC	☑ Project No 1	□ Project No 2	□ Project No 3	0.041667
AAC	☑ Project No 1	□ Project No 2	□ Project No 3	0.125
AMC	☑ Project No 1	□ Project No 2	□ Project No 3	0.3125
D3C2	☑ Project No 1	□ Project No 2	□ Project No 3	0.225

DYNAMIC METRICS

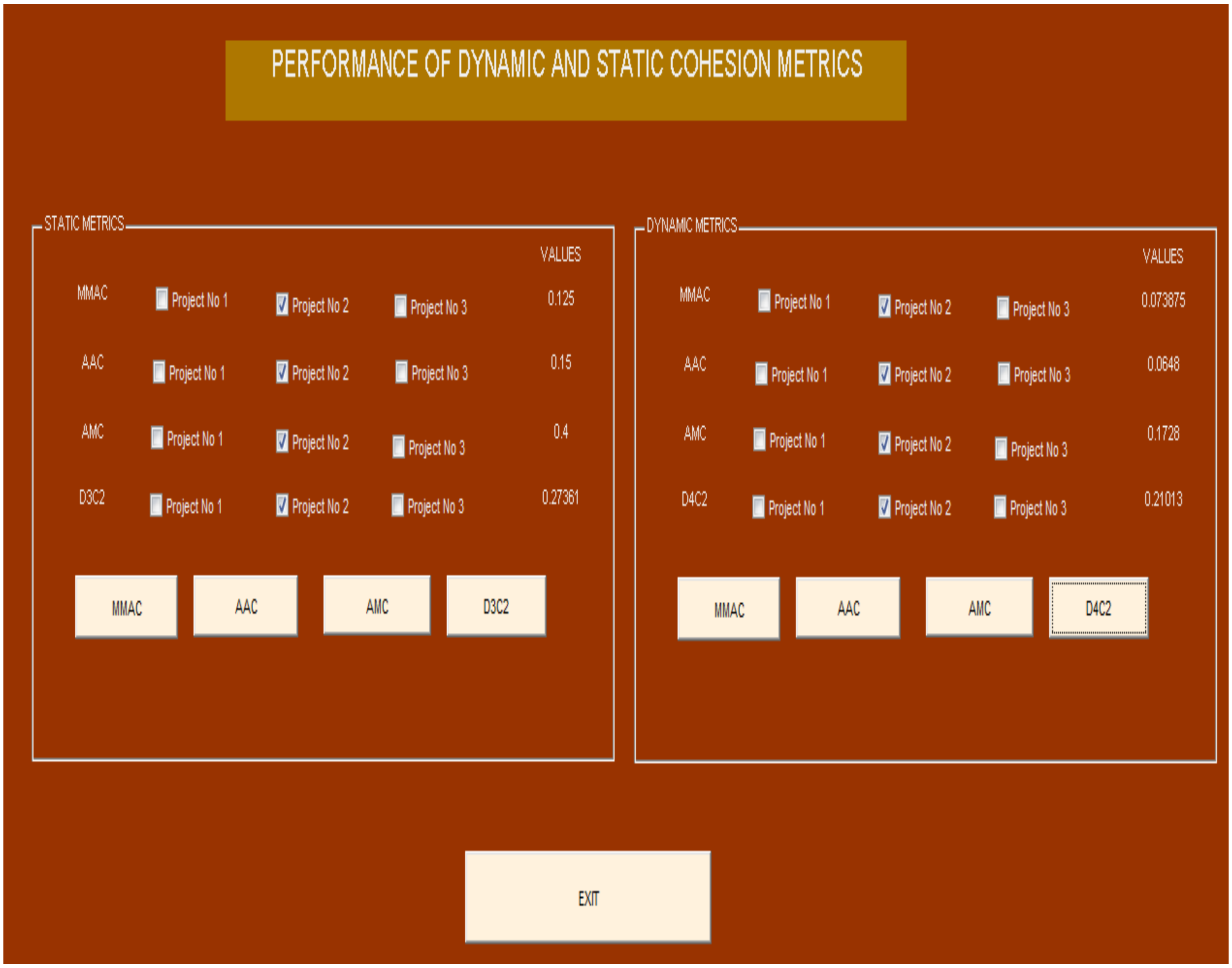
				VALUES
MMAC	☑ Project No 1	□ Project No 2	□ Project No 3	0.018042
AAC	☑ Project No 1	□ Project No 2	□ Project No 3	0.094375
AMC	☑ Project No 1	□ Project No 2	□ Project No 3	0.23594
D4C2	☑ Project No 1	□ Project No 2	□ Project No 3	0.0972

Fig. 4.4 Calculation of static and dynamic cohesion for project 1

- D_3C_2 metric is applying definition use pair dynamic approach on the D3C2 static metric to calculate the cohesion value is dynamically for the project1.
- To select the projec1 and calculate the MMAC, AAC, AMC and D3C2 values.
- To provide the accurate result as compare to the static metric.
- Project1 static is providing less accurate values as compared to the dynamic analysis project1.

Table 4.1 Show static and dynamic cohesion for project 1

	MMAC	AAC	AMC	D3C2
Project1 (Static)	0.041667	0.125	0.3125	0.225
Project 1 (Dynamic)	0.018042	0.094375	0.23594	0.0972



PERFORMANCE OF DYNAMIC AND STATIC COHESION METRICS

STATIC METRICS

				VALUES
MMAC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.125
AAC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.15
AMC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.4
D3C2	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.27361

DYNAMIC METRICS

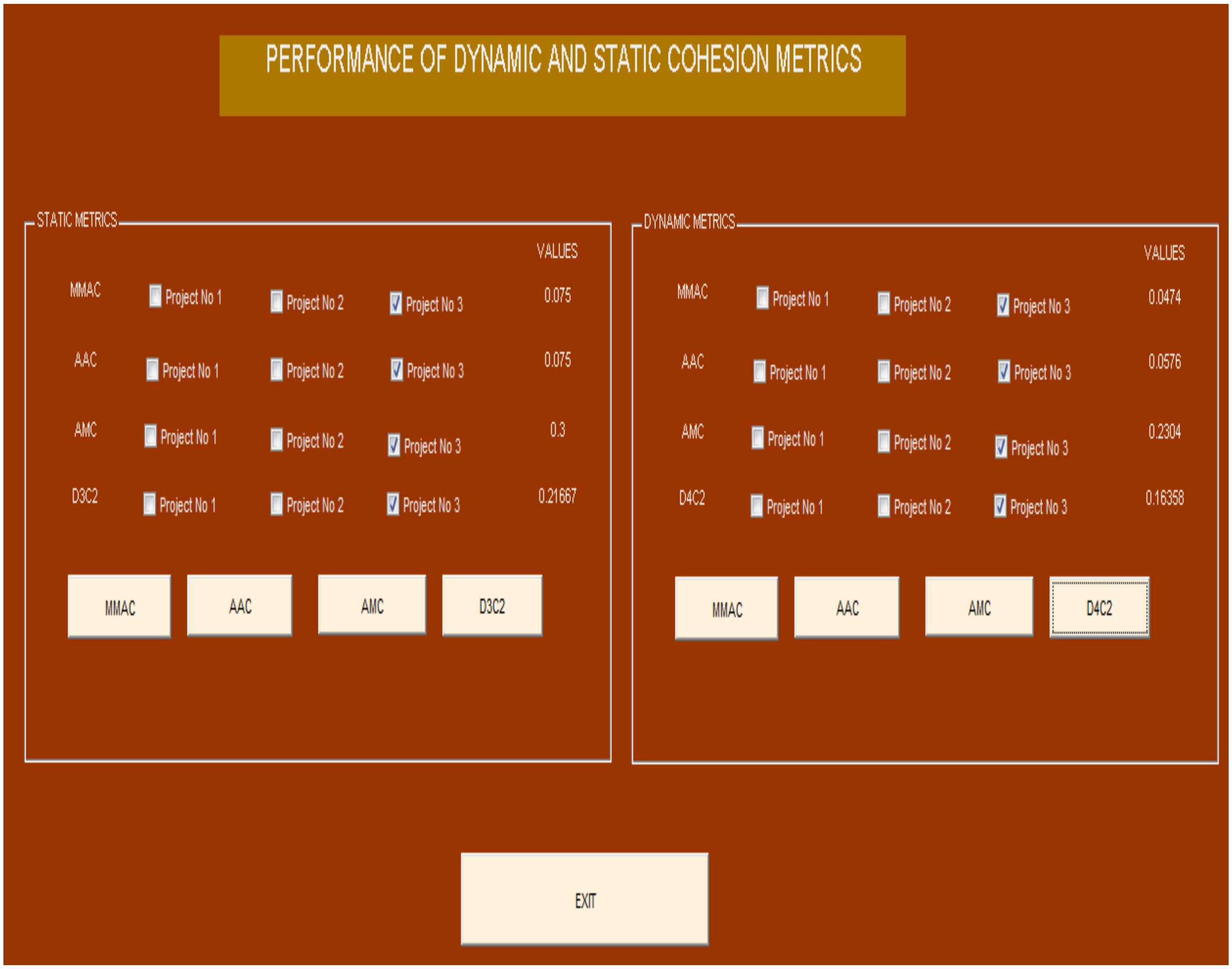
				VALUES
MMAC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.073875
AAC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.0648
AMC	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.1728
D4C2	☐ Project No 1	☑ Project No 2	☐ Project No 3	0.21013

Fig 4.5 Calculation of static and dynamic cohesion for project 2

- ➢ D3C2 metric is applying definition use pair dynamic approach on the D3C2 static metric to calculate the cohesion value is dynamically for the project2.
- ➢ To select the projec2 and calculate the MMAC, AAC, AMC and D3C2 values.
- ➢ To provide the accurate result as compare to the static metric.
- ➢ Project2 static is providing less accurate values as compared to the dynamic analysis project2.

Table 4.2 Show static and dynamic cohesion for project 2

	MMAC	AAC	AMC	D3C2
Project 2 (Static)	0.125	0.15	0.4	0.27361
Project 2 (Dynamic)	0.073875	0.0648	0.1728	0.21013



PERFORMANCE OF DYNAMIC AND STATIC COHESION METRICS

STATIC METRICS

VALUES

MMAC □ Project No 1 □ Project No 2 ☑ Project No 3 0.075

AAC □ Project No 1 □ Project No 2 ☑ Project No 3 0.075

AMC □ Project No 1 □ Project No 2 ☑ Project No 3 0.3

D3C2 □ Project No 1 □ Project No 2 ☑ Project No 3 0.21667

DYNAMIC METRICS

VALUES

MMAC □ Project No 1 □ Project No 2 ☑ Project No 3 0.0474

AAC □ Project No 1 □ Project No 2 ☑ Project No 3 0.0576

AMC □ Project No 1 □ Project No 2 ☑ Project No 3 0.2304

D4C2 □ Project No 1 □ Project No 2 ☑ Project No 3 0.16358

Fig 4.6 Calculation of static and dynamic cohesion for project 3

- D3C2 metric is applying definition use pair dynamic approach on the D3C2 static metric to calculate the cohesion value is dynamically for the project3.

Table 4.3 Show static and dynamic cohesion for project 3

	MMAC	AAC	AMC	D3C2
Project 3 (Static)	0.075	0.075	0.3	0.21667
Project 3 (Dynamic)	0.0474	0.0576	0.2304	0.16358

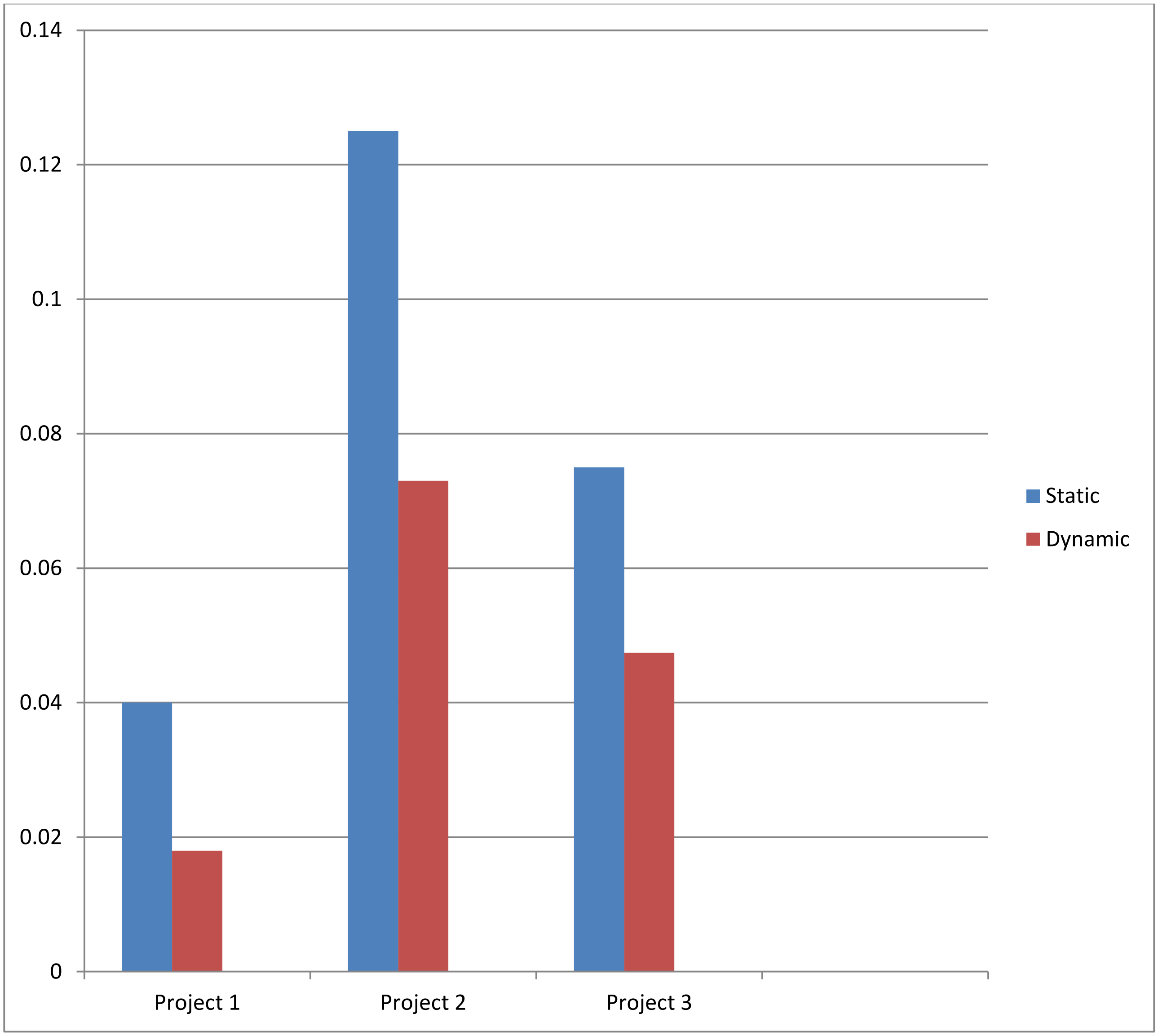


Fig 4.7 Bar chart for Static and Dynamic comparison of MMAC

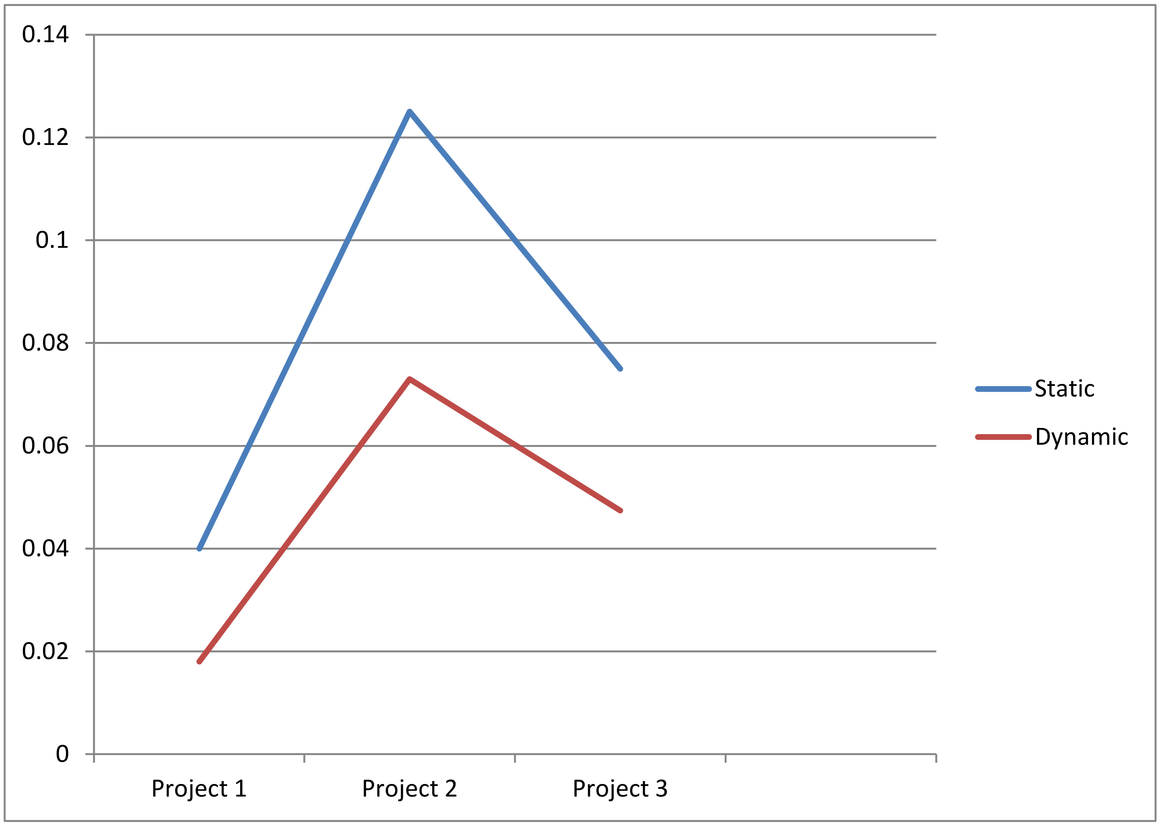


Fig 4.8 Line graph for Static and Dynamic comparison of MMAC

- D_3C_2 is the static metric used to calculate the cohesion value statically.
- Using Definition use-pair dynamic approach on the D_3C_2 to calculate MMAC cohesion values dynamically.

Table 4.4 Show Static and Dynamic comparison of MMAC

	Project1	Project2	Project3
Static MMAC	0.04	0.125	0.075
Dynamic MMAC	0.0180	0.0738	0.0474

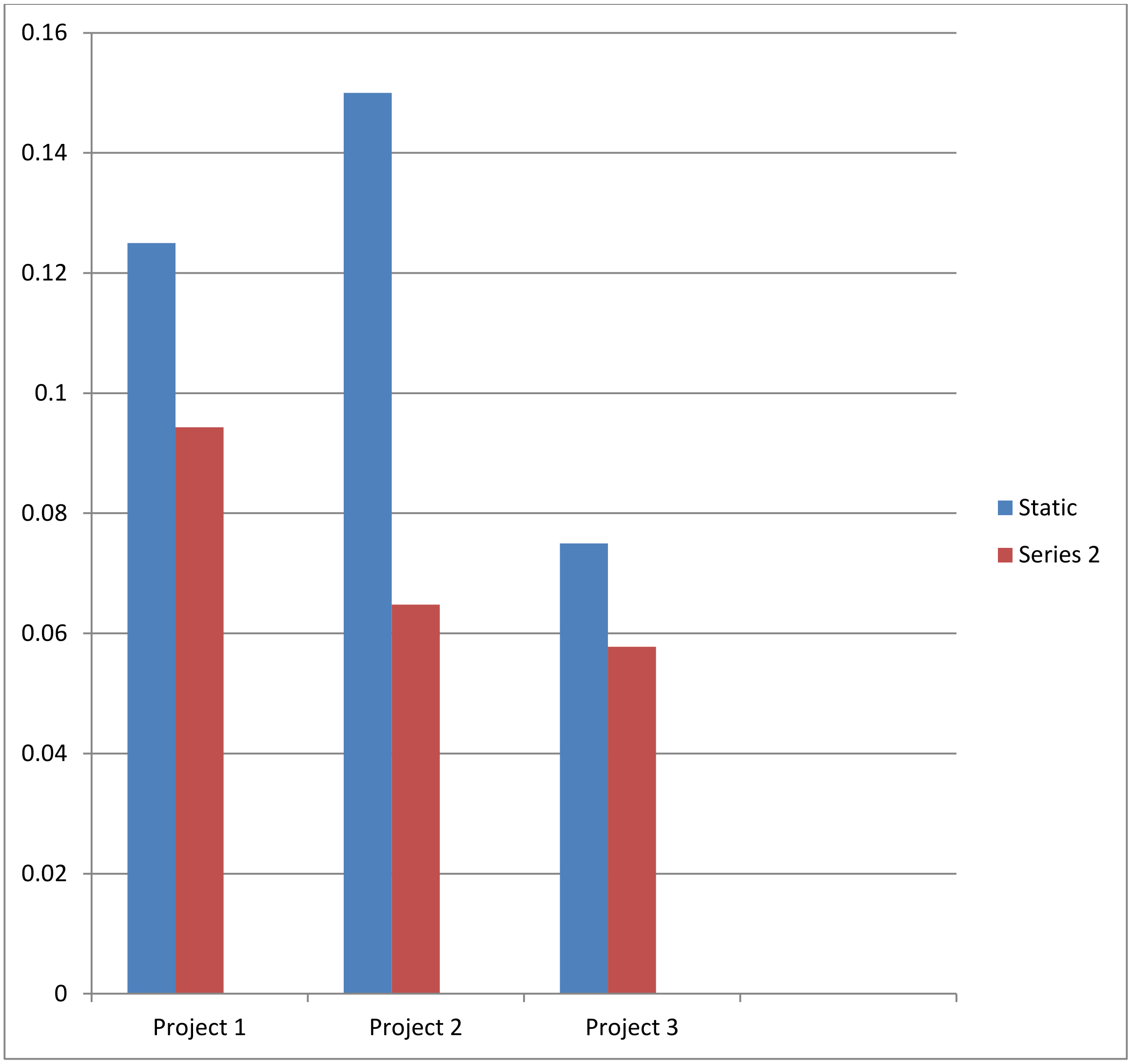


Fig 4.9 Bar chart for Static and Dynamic comparison of AAC

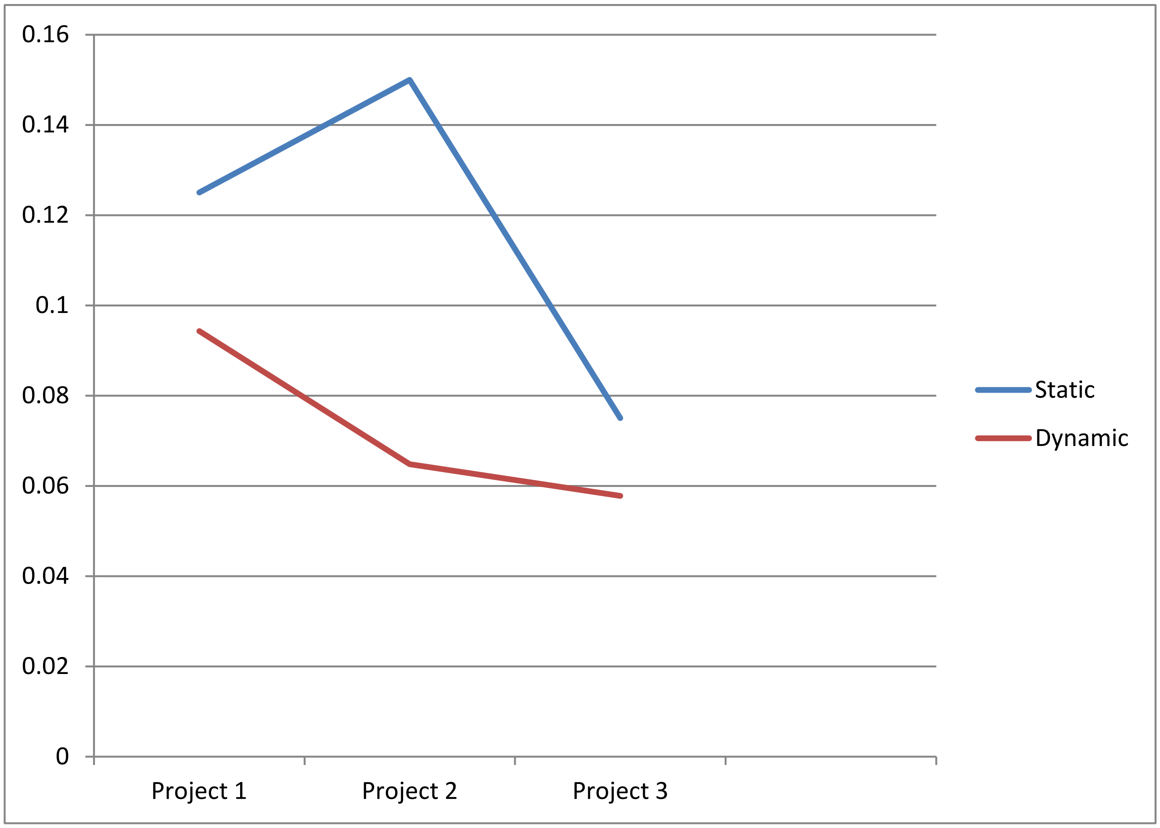


Fig 4.10 Line Graph for Static and Dynamic comparison of AAC

D_3C_2 is the static metric used to calculate the cohesion value statically.

- Dynamic slicing approach is applied on the D_3C_2 to calculate AAC (Attribute-attribute cohesion) is dynamically.

Table 4.5 Show Static and Dynamic comparison of AAC

	Project1	Project2	Project3
Static AAC	0.125	0.15	0.075
Dynamic AAC	0.0943	0.0648	0.0578

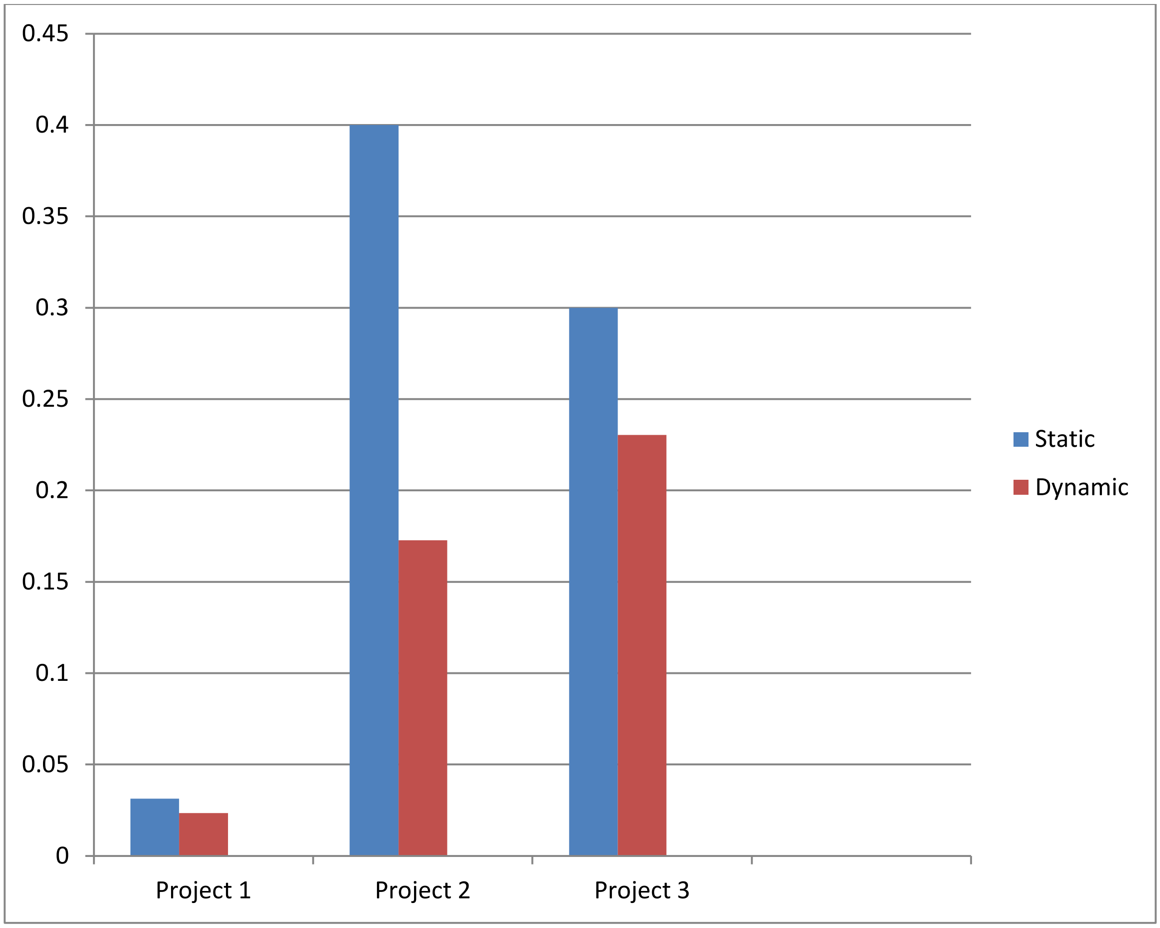


Fig 4.11 Bar chart for Static and Dynamic comparison of AMC

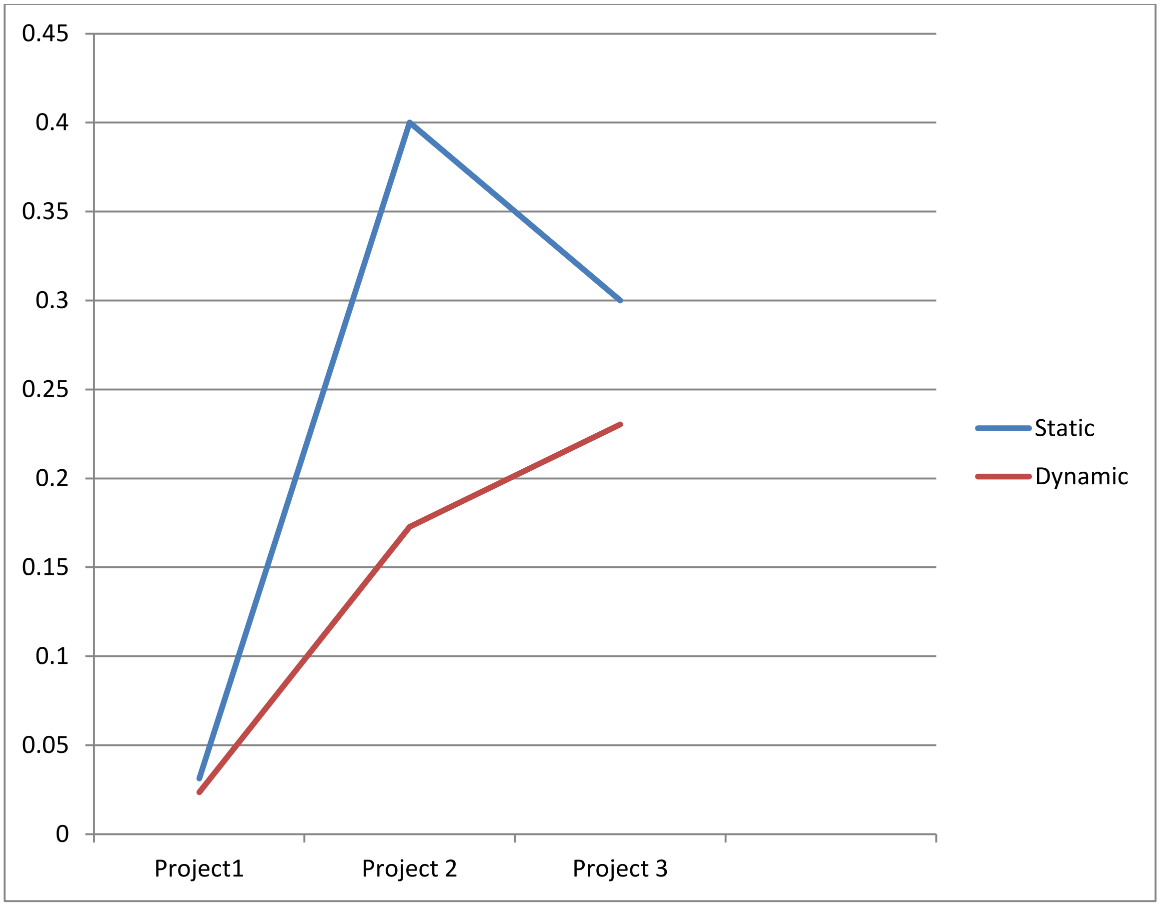


Fig 4.12 Line graph for Static and Dynamic comparison of AMC

- D_3C_2 is the static metric used to calculate the cohesion value statically.
- Dynamic slicing approach is apply on the D_3C_2 to calculate AMC (Attribute-Method cohesion) is dynamically

Table 4.6 Show Static and Dynamic comparison of AMC

	Project1	Project2	Project3
Static AMC	0.03125	0.4	0.3
Dynamic AMC	0.0235	0.1728	0.2304

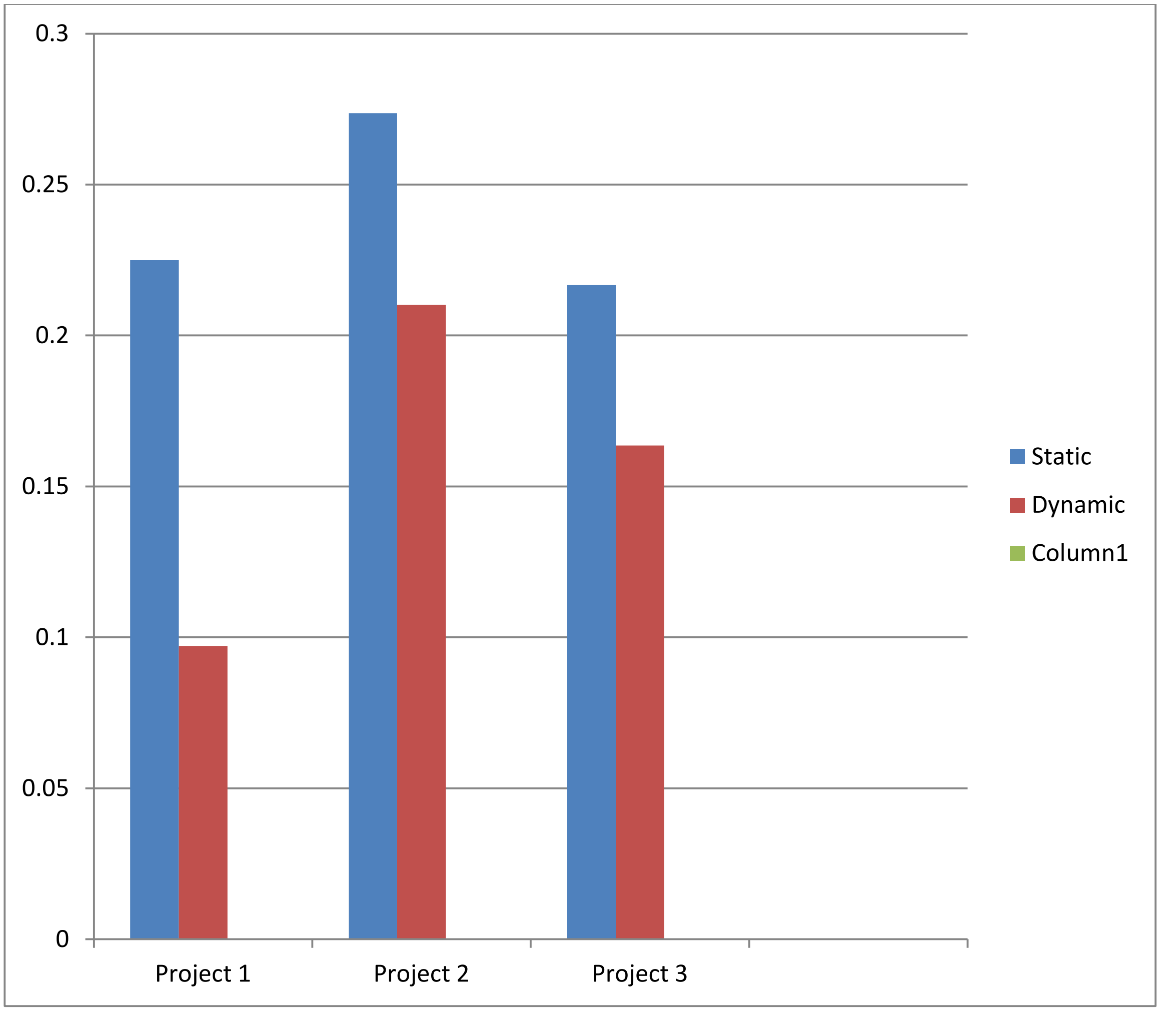


Fig 4.13 Bar chart for Static and Dynamic comparison of D_3C_2

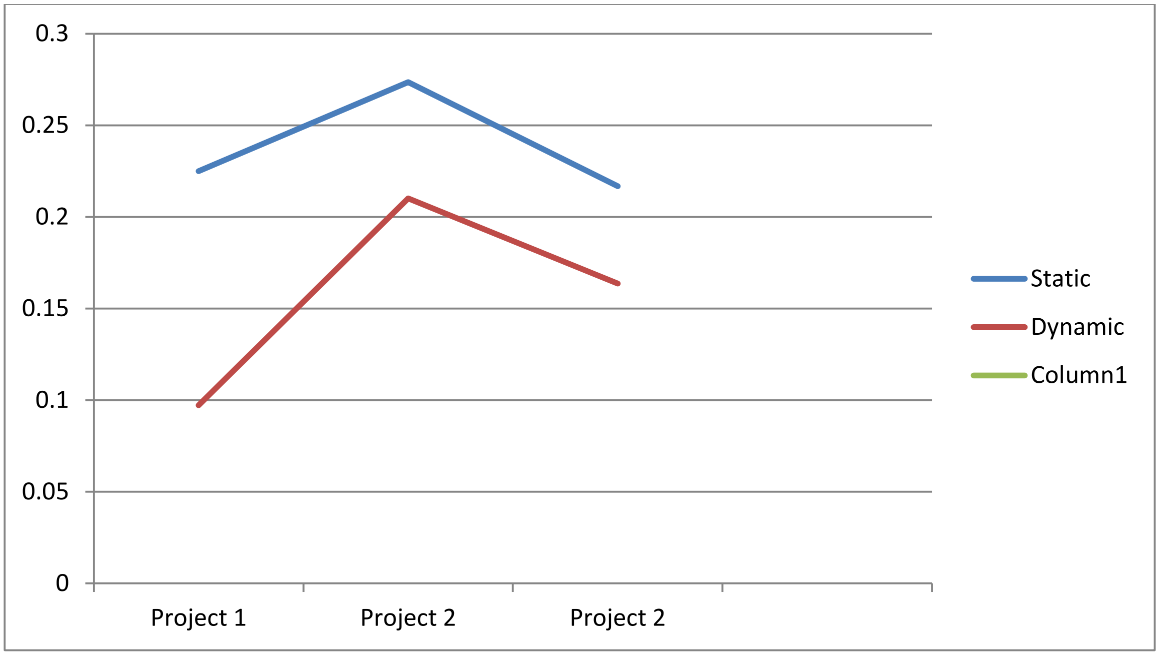


Fig 4.14 Line graph for Static and Dynamic comparison of D_3C_2

- D_3C_2 is the static metric used to calculate the cohesion value statically.
- Dynamic slicing approach is applied on the D_3C_2 to calculate cohesion is dynamically.

Table 4.7 Show Static and Dynamic comparison of D_3C_2

	Project1	Project2	Project3
Static Cohesion values of D_3C_2	0.225	0.27361	0.21667
Dynamic Cohesion values of D_3C_2 i.e. D_4C_2	0.0972	0.21013	0.16358

TOOL USED: MATLAB

MATLAB stands for **“Matrix Laboratory”**. MATLAB is a high level language. It is a program for doing numerical computation, analyzing images and data. It was originally designed for solving linear algebra type problems using matrices. Its name is derived from Matrix Laboratory. It helps us in solving the problems faster than other languages and used in various applications such as signal processing, image processing, communications, computational biology and control design. MATLAB system has following these parts:

- Desktop tools and development environment
- Mathematical function library
- The language
- Graphics
- External interfaces

FEATURES OF MATLAB:

- Environment for managing the code, files and data.
- 2-D and 3-D graphics functions for analyzing the data.
- Provide interactive tools for solving problems.
- Provide functions for integrating the MATLAB based algorithms with external applications.
- High level language.
- It helps in solving the problems faster than other languages.

STANDARD WINDOWS IN MATLAB:

- **Command Window**: - The window where you type and execute commands.
- **Workspace Window**: - This shows current variables and allows to edit variables by opening array editor (double click), to load variables from files and to clear variables.
- **Current Directory window**: - this shows current directory and MATLAB files in current folder, provides with a handy way to change folders and to load files.

- **History window**: - This shows previously executed commands. We can re-execute the Commands by double-clicking

MATLAB HELP:

- Help option is present on the top of the window in the right side.
- MATLAB help is a powerful way for learning the MATLAB.
- It not only contains the theoretical background, but also shows demos for implementation.
- We can search any command by typing in the search box.
- It explains the commands searched by you with examples

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

The problem to improve the quality of cohesion metrics led to emergence of dynamic analysis. The proposed dynamic cohesion measures are more accurate it because take into consideration run-time behavior of classes. The proposed dynamic technique uses def-use pairs have been applied on the static metric D_3C_2 to analyze it dynamically. Using dynamic techniques, we can obtain more accurate measurement of functional cohesion compared with new techniques. The static cohesion metric is using some dynamic approaches to obtain accurate results and better considerations of run time behavior then the existing metrics. Definition use pair is dynamic technique using dynamic slicing criteria on the some static metrics to take more accuracy. The proposed dynamic cohesion measures are better indicators of external software quality attributes such as change defects than the existing cohesion metrics. To use any other dynamic approaches to perform dynamic analysis and to take better and more accurate cohesion values in future. The introduced metric can be improved in several directions such as dynamic analysis. In Future, use another static metric performs dynamic analysis to carry out the cohesion values are more accurate.

CHAPTER 6
REFERENCES

Robert s. Arnold, *"Software Reengineering."* IEEE computer society los, California.
Nasib Singh Sill, *"Software Reengineering"*
Rogers. Pressman (PhD) "Software engineering.

[1] Nguyen Van Vy. Nguyen Vietnam – curriculum software engineering, *IT Facility – Hanoi University of Technology*, VNU, 2006.

[2] Jehad Al Dallal- A Design-Based Cohesion Metric for Object- Oriented Classes- *World Academy of Science, Engineering and Technology International Journal of Computer, Information, Systems and Control Engineering,* 2007.

[3] Payal Khurana & Puneet Jai Kaur- Dynamic metric at design Level- *International Journal of Information Technology and Knowledge Management* July-December 2009, Volume 2, No. 2, pp. 449-454.

[4] Kuljit Kaur and Hardeep Singh-An Investigation of Design Level Class Cohesion Metrics-*Department of Computer Science and Engineering, Guru Nanak Dev University, India* January 2012.

[5] N.Sasirekha, A.Edwin Robertand Dr.M.Hemalatha-Program slicing techniques and its applications -International Journal of Software Engineering & Applications (IJSEA), *Vol.2, No.3, July 2011.*

[6] N. Gupta, P. Rao, Program execution based module cohesion measurement, *in: Proc. 16th International Conference on Automated on Software Engineering (ASE '01), San Diego, USA, 2001.*

[7] A. Mitchell, J.F. Power, Run-Time Cohesion Metrics for the Analysis of Java Programs, *Technical Report Series No. NUIM-CS-TR-2003-08, National University of Ireland, Maynooth Co. Kildare*, *Ireland,* 2003.

[8] Esubalew Alemneh-Current States of Aspect Oriented Programming Metrics-***Volume*** *3 Issue 1, January* 2014.

[9] Varun Gupta, Jitender kumar Chhabra, dynamic cohesion measures for object-oriented software-*Journal of system architecture* (2011).

[10] Jitender Kumar Chhabra and Varun Gupta-A Survey of Dynamic Software Metrics-*Journal of computer science and technology*(2010).

[11] Payal Khurana & Puneet Jai Kaur-Dynamic Metrics At Design Level-*International journal of information technology and knowledge management(2009).*

[12] Shweta Sharma & Dr.s.srinivasan-A Review of Coupling and Cohesion Metrics in Object Oriented Environment- *Journal of computer science and technology(2013).*

[13] Cholla Bonja, K Ebob, “Metrics of class cohesion Similarity between Methods”-

Proceedings of the 44th annual Southeast regional conference

[14] Marcus, A.; Poshyvanyk, D., "The conceptual cohesion of classes," *Software Maintenance, 2005. ICSM'05. Proceedings of the 21st IEEE International Conference on*,vol.,no.pp.133,142,Sept.(2005)

[15] Dr.Linda H.Rosenbar “*By software assurance technology centre”, Software assurance Technology centre of Unisys federal System, 2010*

I http:/journals.ecs.soton.ac.uk/tutorial/java/objects/object.htm

II http:/softwaredesign.com/objevts.html

III http:/ www.infoq.com/articles/projectmetrics

APPENDIX

Abbreviations:

RCI	-	Ratio of Cohesion interaction
DCC	-	Dynamic Class Cohesion
NHD	-	Normalized Hamming Distance
CAMC	-	Cohesion among Method of Class
SNHD	-	Scaled Normalized Hamming Distance
D_3C_2	-	The Distance Design-Based Direct Class Cohesion
D_4C_2	-	The Dynamic Distance Design-based Direct Class Cohesion