Optimization of Reusability model for Aspect Oriented Software Development (AOSD)

Submitted By:

Muhammad Yasir Ejaz

276-FBAS/MSSE/F09

Supervisor:

Dr.Abdul Rauf

Department of Computer Science & Software Engineering

Faculty of Basic and Applied Sciences

International Islamic University Islamabad

(2012)

In the name of ALLAH most beneficent,
ever merciful

Department of Computer Science and Software Engineering

International Islamic University Islamabad

Date: 15-08-2012

FINAL APPROVAL

This is to certify that we have read the thesis submitted by **Muhammad Yasir Ejaz**, Reg. # **276-FBAS/MSSE/F09**. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of MSSE.

Committee:

Supervisor:

Dr.Abdul Rauf

Assistant Professor,

Department of Computer Science and Software Engineering,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad

Internl Examiner:

Syed Muhammad Saqlain

Assistant Professor,

Department of Computer Science and Software Engineering,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad

External Examiner:

Dr.Muhammad Ramzan

Assistant Professor,

Department of Computer Science

Foundation University, Rawalpindi

DEDICATION

I would like to dedicate my research work to the

HOLIEST man Ever Born on Earth,

PROPHET MUHAMMAD (Peace Be Upon Him)

and

I also dedicate my work to my

PARENTS

Whose sincere love and prayers were a source of strength for me and made me to do this research work successfully.

Muhammad Yasir Ejaz
276-FBAS/MSSE/F09

A dissertation submitted to the

Department of Computer Science & Software Engineering,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad,

as a partial fulfillment of the requirements

for the award of the degree of

MS in Software Engineering (MSSE)

DECLARATION

I hereby declare that this Thesis ***"Optimization of Reusability model for Aspect Oriented Software Development (AOSD)"***, neither as a whole nor as a part thereof, has been copied out from any source. It is further declared that I have written this thesis entirely on the basis of my personal efforts, made under the proficient guidance of my thesis supervisor, **Dr. Abdul Rauf**

If any part of this research thesis proved to be copied or found to be a research of some other individual, I shall standby the consequences.

No portion of the research work presented in this thesis report has been submitted in support of any other degree or qualification of this or any other University or institute of learning.

Muhammad Yasir Ejaz
276-FBAS/MSSE/F09

Project In Brief

Project Title:	Optimization of Reusability model for Aspect Oriented Software Development (AOSD)
Undertaken By:	Muhammad Yasir Ejaz
Supervised By:	Dr.Abdul Rauf
Start Date:	August 2011
End Date:	June 2012
Tools & Technologies	MS Office 2007, Weka, MS Visual Studio 2008
Documentation Tool:	MS Office 2007
Operating System:	Windows 7

Abstract

Due to increasing complexity and ever changing demands and customization of software system, reusability is becoming most important quality attribute . Reusability reduces the development time and cost of the development and decrease the effort required for software testing as well.

On the other hand Aspect Oriented Software Development (AOSD) is a modular approach focusing on cross cutting concerns and dealing with the issues like code scattering and tangling found in other software development techniques.

Reusability has been addresses by various software development models and different models have been proposed to accommodate reusability requirements but with respect to AOSD, reusability has not been addressed to the extent that it deserves considering the increased role of reusability in software development.

In this research effort , We have evaluated different reusability models with respect to Aspect Oriented Software Development list down the factors that can helpful in ensuring reusability in AOSD. We have used different techniques to analyze the available models with AOSD and measured the extent to which they are capable of achieving the goal of reusability.

Finally applied Fuzzy logic to our survey results to get the maximum reusability depending on different factors.

Table of Contents

Chapter: 1. Introduction 11

1- Introduction: 12

1.1- Aspect Oriented Software Development 12

1.2- Software Reusability 12

1.3- Related Work: 14

1.4- Problem Statement 15

1.5- Research Questions and Objectives 15

1.6- Research methodology 16

1.7- Thesis Organization 17

Chapter: 2. Literature Review 18

2- Literature Review 19

2.1- Reusability and AOSD 19

2.2- Object Oriented and Aspect Oriented 20

2.3- Reusability Assessment Models 21

2.4- Reusability Attribute Model 23

Chapter: 3. Industrial Survey 25

3- Industrial Survey 26

3.1- Introduction 26

3.2- Results 27

Chapter: 4. Survey Results and Fuzzy Logic 37

4- Survey Result Analysis 38

4.1 Fuzzification of results 39

4.2 Comparison of Fuzzy Result with C4.5 Algorithm 54

5- Final Results 60

6- Conclusion and Future Work 68

6.1 Conclusion: 68

6.2 Recommendations for Future Work 68

7- References 69

Chapter: 1. Introduction

1- Introduction:

In this section we will define AOSD, reusability, software product line, different models of software reusability in traditional software development and Aspect Oriented Software Development. We will also discuss different model of Reusability with respect to AOSD

1.1- Aspect Oriented Software Development

A traditional Software system is comprised of several cross cutting concerns or aspect more precisely. In current software development methodologies, code scattering and tangling are two major difficulties effecting development in the ways of poor tractability, lower productivity, less code reuse and poor code quality. Aspect Oriented Software Development (AOSD) has modularity of concerns and has solved the above mentioned issues. AOSD focuses on modularization of cross cutting concerns. The modular unit in AOSD in Aspect. Aspect Oriented development is achieved in three steps involving aspectual decomposition, concern implementation and finally aspectual recomposition. Aspect Oriented Software Development implements individual concerns in a loosely coupled fashion, and combine these implementations to form the final system. [1][2][3]

1.2- Software Reusability

Reusability of the software systems is becoming a very important factor due to rapid software development and increasing complexity. Software reusability improves the quality of software product by reducing development time, effort and cost. By applying an aspect-oriented approach, such concerns can be isolated resulting in the increase maintainability and reusability of the system.[15][16][19]

Increasing demand of software products and day to day customization has increased the more and improved quality of software and software development techniques. Different quality models have been proposed by researcher to accommodate different quality attributes such as

maintainability, usability, efficiency, performance, overall functionality, reliability, portability and reusability of code components. [15].Many approaches such as Module-Oriented (MO), Object-Oriented (OO), and Component-Based Software Development (CBSD) are used to develop reusable software[19]. In OO technology, the code related to a concern scattered in multiple classes, affect reusability and maintainability. These scattered concerns are called crosscutting concern. Separation of these concerns (AOSD) help in improvement of reusability and maintainability of software components. [2]

Different models have been proposed by researchers using AOSD technique based on quality attributes of the Software Systems[15][16][17][19]. Aspect Oriented Software Development is an emerging area and practical implementation of the proposed model has not been achieved.

Aspect Oriented Software development is based on the concept of separation of concerns and each concern is considered as an aspect. Reusability models have focused on different factors enhancing reusability goal [16][17]. These models are intended to access the reusability of software components. Assessment of Aspect Oriented Software model is different from traditional software development. We will evaluate how far those models have been useful to achieve the desired goals of reusability.

Reusability has been addresses in Aspect oriented software development models but it is based on the other quality attributes such as flexibility, maintainability, portability, scope coverage understandability and many more. [15].

AOSD is an emerging area model has been proposed for AO development but mainly the focus is on separation of concerns and issues in other software development techniques. As AOSD is a modular approach having the concept of separation of cross-cutting concerns, so reusability as a separate concern can be addresses in AOSD.

Mainly the requirements of reusability are ambiguous or unknown and not addressed in software development model.

Researchers have proposed different reusability model using AOSD technique based on quality attributes of the Software Systems[15][16][17][19]. Aspect Oriented Software Development is an emerging area and practical implementation of the proposed model has not been achieved.

Aspect Oriented Software development is based on the concept of separation of concerns and each concern is considered as an aspect. Reusability models have focused on different factors enhancing reusability goal [16][17]. These models are intended to access the reusability of software components. Assessment of Aspect Oriented Software model is different from traditional software development. We will evaluate how far those models have been useful to achieve the desired goals of reusability

1.3- Related Work:

Different models have been proposed by researchers using AOSD technique based on quality attributes of the Software Systems[15][16][17][19]. Aspect Oriented Software Development is an emerging area and practical implementation of the proposed model has not been achieved.

Aspect Oriented Software development is based on the concept of separation of concerns and each concern is considered as an aspect. Reusability models have focused on different factors enhancing reusability goal [16][17]. These models are intended to access the reusability of software components. Assessment of Aspect Oriented Software model is different from traditional software development. We will evaluate how far those models have been useful to achieve the desired goals of reusability.

Reusability is more important for software product line where software component even modules are to be used in diversity of products.

Literature purposed different models for reusability based on different functional and non functional attributes.

1.4- Problem Statement

In introduction section and related work I have mentioned that there are already many proposed model of software reusability but reusability is not addressed with respect to Aspect Oriented Software Development being a cross-cutting concerns and dependent on other aspect. AOSD is not commonly used and comparatively new area than others. Somehow reusability is addressed with respect to AOSD but not in the form of a proposed model.

My ambition for this research is to purpose such model and to test its result performing some experiments and improving reusability with respect to AOSD.

We would be focusing on software product line for experiment so reusability with its purpose can be addressed.

1.5- Research Questions and Objectives

Researchers have proposed different reusability model using AOSD technique based on quality attributes of the Software Systems[15][16][17][19]. Aspect Oriented Software Development is an emerging area and practical implementation of the proposed model has not been achieved. Aspect Oriented Software development is based on the concept of separation of concerns and each concern is considered as an aspect. Reusability models have focused on different factors enhancing reusability goal [16][17]. These models are intended to access the reusability of software components. Assessment of Aspect Oriented Software model is different from traditional software development. We will evaluate how far those models have been useful to achieve the desired goals of reusability.

Fuzzy Logic is a mathematic technique to reach a definite conclusion on the basis of incomplete, ambiguous, vague or improper data. [19] We will use different quality factor of software systems as an input for using Fuzzy Logic method to get an optimized model of the reusability with respect to AOSD. The input data would be based on the software industry survey using AOSD to achieve the goals of reusability. This technique will help in selection of the best model in term of reusability.

Question 1: What is the state of art of Reusability w.r.t current AOSD Models and state of practice of reusability w.r.t. Local Market? [15][16][17][19]

Question 2: How can we use Fuzzy Logic to optimize reusability models to overcome the issues in existing AOSD with respect to reusability?

1.6- Research methodology

To achieve the given research goals we will conduct a vast literature review based on different journal, conference proceeding, digital libraries and search engines, to find out the most relevant research by using proper search string. For identification So far used literature resources are as follow. A range of data bases has been selected for rigorous search and focusing on the topics. We will use IEEE Explore, ACM Digital library, Google scholar (scholar.google.com), Cite Seer library, Science Direct and other available resources to get the answers of research questions. We have divided research methodology into following steps:

1st Step: To study and analyze the already proposed reusability model a literature review would be conducted. We are not performing a systematic literature review but the purpose is to find out the success factor of existing AOSD models focusing on reusability. We will follow the guidelines of B. A. Kitchenham to take help to conduct a literature review.[22] For literature review, we have used different resources including journal, conference proceeding, digital libraries and search engines. For identification of primary studies of systematic review a clearly search strategy and quality assessment criteria has been defined

2nd Step: Prioritizing reusability models on the basis of their focus to achieve the maximum goal of reusability.

3rd Step: Survey will be conducted to collect the data from organization to sort out the techniques used to achieve the reusability of software systems or software components.

The purpose of this survey is to identify the models or software development techniques in the software industry of Pakistan and how needs of software reusability are addressed by

following those models or techniques. Models and needs would be prioritized on the basis of the feedback. The survey is focused on the software development industry of Pakistan.

4th Step: Comparison of the reusability models with the market requirements.

5th Step: Fuzzy logic implementation to optimize reusability model to meet the market requirements with respect to AOSD.

6th Step: Separation of key findings and if there is space for future work.

7th Step: Finally recommendation and conclusion work will be documented.

1.7- Thesis Organization

This thesis consists of 6 chapters. The chapter 1 is the introduction in which the motivation, problem statement and research question and objective has to be covered. After introduction in chapter 1, Chapter 2 will cover the Literature Review of software reusability and Aspect Oriented Software Development and study of different models is done. Chapter 3 focuses on the results of the survey which has been conducted. In chapter 4, we have discussed the analysis of the survey results using different techniques. In chapter 5 we have applied our results to the case study. Chapter 6 a brief conclusion and future work has been presented.

Chapter: 2. Literature Review

2- Literature Review

In this section we have discussed how software reusability is important and how software reusability is covered in AOSD and what are different model of software reusability with respect to AOSD and without AOSD. We have selected on model in those section that defines the reusability dependency on different attributes and we will use those attributes to achieve the reusability to its maximum level.

2.1- Reusability and AOSD

Typically a software system comprise of several crosscutting concerns that can be distinguished in separate Aspects. Software design and development is affected in many ways by the problem arising due to code tangling and code scattering.

Aspect oriented software development allows the modularizing of the concerns in separates resulting into cross cutting resolving the problem arise because of above mentioned difficulties.[1]

In the past few decades AOSD is becoming a technique to modularize the cross cutting concerns in a software system and resulting into better and improved software development and a better return on investment (ROI).

Many frameworks like JBoss, AspectJ and many more already exist for AOP. IBM is using AOSD for IBM WebSphere Application Server.

The selection of AOSD is dependent on the type of the project. In some case modularity may not result into expected results.[11]

Identification of Aspects during the early phase of life cycle helps in leading advents of AOSD.

AOP results in clean, well encapsulated objects by explicit separation of concerns that tends to be cross through overall system.

AOSD is implemented at the design and programming using Aspect oriented design and Aspect Oriented Programming (AOP). Aspect oriented design focus on the separation of concerns.

Better Aspect Oriented Design results in better development phase. The modular unit is called Aspect. The whole process of separation of concerns can be divided into three phase starting from decomposition, aspect implementation and finally conceptual composition of the system. Better approach in these steps results in better management and less complexity of code.

From the literature review we have found that there many researchers have proposed theoretically model to assess the reusability and especially with respect to Aspect Oriented Software Development.

2.2- Object Oriented and Aspect Oriented

Reusability in Object Oriented system can be measured using different metrics and models.[1,13,14,15,16,17,18-19]

But due to modularization of cross cutting concerns it is not possible to apply those techniques in AOSD as well. Code characteristics differ in AOSD than OO.

Basically reusability is the degree of a module to which it can be reused over the period of time in different systems with or without modifications.

Mainly in AO reusability is dependent on some non functional chrematistics like modularity, understandability, adaptability and many more. OO metrics and model cannot be applied to AO concept because the type of abstraction differs in AO and OO.

An empirical study has proposed ten different quality characteristics on which assessment of reusability can be performed. All these attributes may not be used in assessment. They include understandability, modularity, portability, completeness, adaptability, flexibility, expandability, efficiency, expandability and correctness. [13-19]

The degree of dependency of reusability on these attributes can be assessed by conducting a survey from local software market. This includes the second phase of our research methodology.

In context of software product line the degree of reusability of a component depends upon the requirements and scope of the system.

2.3- Reusability Assessment Models

One of the quality model proposed by Mc Call defines the relationship between simplicity, generality modularity and reusability. [10-15]

Different model and metrics has discussed dependency of reusability on different attributes on whole basis or individual attributes.

A metrics suite for measuring reusability of software components

-Reusability of Black Box components has been addresses in a model in [12-15] on the basis of understandability, adaptability and portability

Commercial off-the-shelf component quality model proposal

-Quality model for components off the shelf (COTS) is addressed in [13-15] dependent on generality and localibility.

On the reuse and maintenance of aspect-oriented software: An assessment framework

-A framework to assess the reusability is presented in [14-15] dependent on reusability and maintainability and sub-factors including flexibility and understandability for assessment.

A framework for evaluating reusability of core asset in product line engineering

-For product line core assets, a framework is proposed in [15-15] is presented on seven attributes. Five out of them are marked as core and remaining two as auxiliary attributes. Core attributes include functional commonality, non-functional commonality, variability richness, applicability and tailor-ability. The other two attributes are understandability and component replacibility. Metrics are provided in research to measure these attributes.

A Proposed Reusability Attribute Model for Aspect Oriented Software Product Line Components

We have selected an approach presented in [15] in which, on the basis of Mc Call model, a reusability model is presented. We have extended our study by conducting market survey on the significance of these attributes and then comparing the results of the model and market survey and putting them together in a case study for results verifications.

For a system or components of a system to be reusable they must has to be:

Easily modifiable: *Modifiability*

Support variability: V*ariability*

Easily understandable: *Understandability*

Flexible for integration: Portability

Common features: *Scope coverage*

Easy error detection and debugging: *Understandability*

Furthermore we can define our attributes as follow

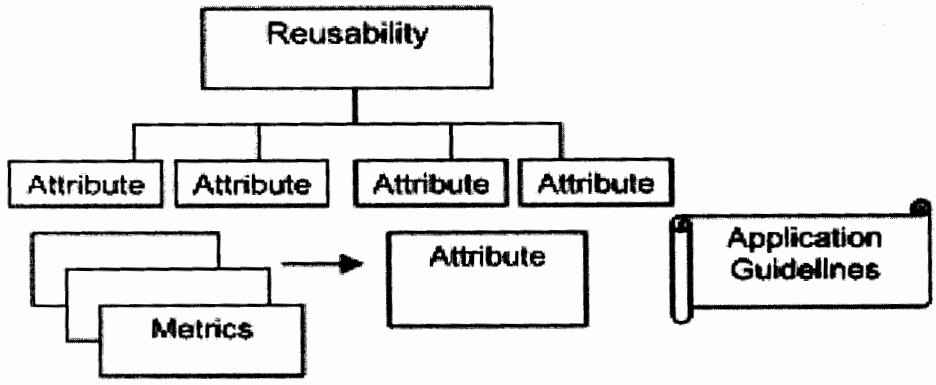


Fig. 1

2.3.1 Flexibility

Effort required to modifying and operational program [21].

2.3.2 Maintainability

The capability of the software product to be modified. Modification may include corrections, improvements or adaptation of the software to change in environment, and in requirements and functional specifications" also defined as "effort required to locate and fix an error in an operational program" [21], [22]. Maintainability can also be defined as the ease with which a component can be maintained between different releases.

2.3.3 Understandability

"The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use" [20]. It is also defined as the "attributes of software that bear on the users' effort for recognizing the logical concept and its applicability" [23].

2.3.4 Variability

"Variability is the degree to which something exists in multiple variants, each having the appropriate capabilities" [19].

2.3.5 Portability

Portability: "The capability of the software product to be transferred from one environment to another" [20]. Or, the "effort required to transfer a program from one hardware configuration and/or software system environment to another" [21], [22]. Or "Portability is the ease with which an application or component can be moved from one environment to another" [21], [22].

2.3.6 Scope coverage

Scope Coverage is the attribute that measures the number of features provided by the component against the total number of features in product line scope.

2.4- Reusability Attribute Model

The model we have selected for our study is based on above attributes like portability, maintainability, understandability, scope coverage and flexibility as we have discussed above.

More flexible component are more easily to modify and more easy to reuse. High potential of variability, increase the chances of using a component in variable products. Scope coverage with modification or enhancements increases the chances of reusability in different products. Greater the portability, greater will be its reusability [19]

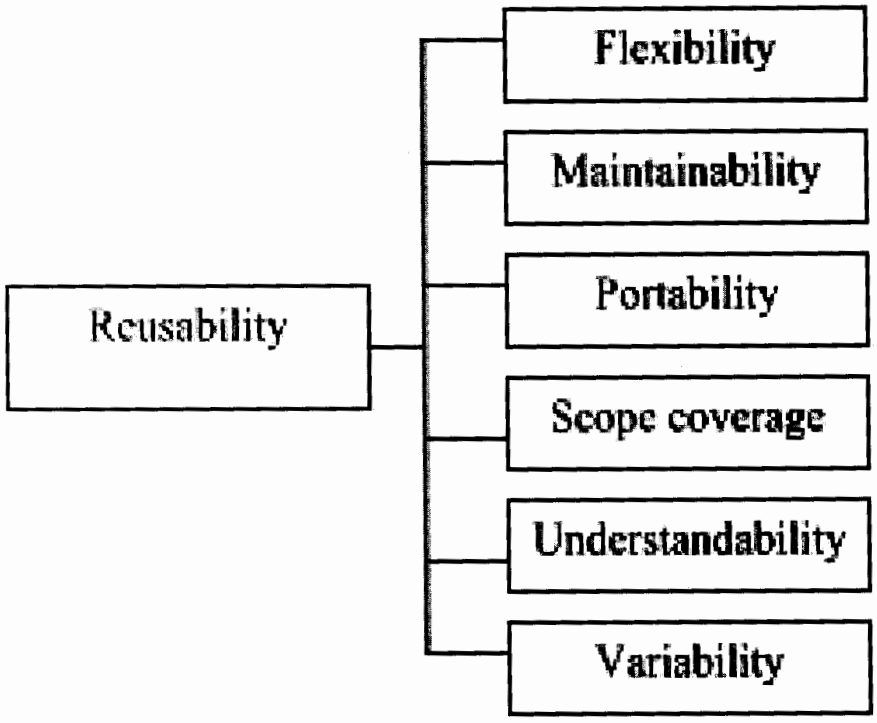


Fig. 2

We will conduct our market survey and further study on the basis of this model and will maximize the reusability depending on these attributes. Our questionnaire will be based on this model and then the next phase our study i.e. fuzzy logic implementation would also be based on these attributes discussed in this proposed model.

Chapter: 3. Industrial Survey

3- Industrial Survey

IN this section we will discuss about the market survey we have conducted on the basis of software reusability model with respect to AOSD. We have divided our survey into section to get the maximum positive output from the market.

3.1- Introduction

In order to evaluate the attribute son which the assessment of reusability is depend with respect to AOSD, a market survey is conducted. Questionnaire of the survey is designed in such a way that first 8 question are about the person and company profile and type of projects they are doing and remaining question were about the quality attributes on the basis of which we can assess the selected model of reusability with respect to AOSD. I have received more than 100 responses from different organization all over the Pakistan. The survey is filled by different person holding different role in their specific organization. We have received response from few very large organizations where number of employees are more than 1000 and working on diversity of projects and product line as well. It is asked in the questionnaire that weather is there any technique used in software development life cycles and more than 50% of the results are in positive. Mostly Agile software development process is practiced in organizations in Pakistan.

In the second section of the survey user were asked indirectly about the use of quality attributes of reusability assessments and reusability practices and techniques used to achieve the goals of reusability.

User of the survey are given choice ranging from strongly disagree to strong agree and in the back ground marking them from 1-5 in range.

Question about the need of reusability and project based on software product line are also asked in the survey to get the clear picture about the practices in the market and make some space to adopt the proposed model to achieve the goals of reusability.

Survey results also helped to understand the importance of reusability and its actual demand in the software industry of Pakistan. Moreover wherever modular approach is used as it was also asked in the survey, implementation of the study results is easy.

3.2- Results

3.2.1- Number of response

Survey is filled by more than 100 users but finally results are filtered and 100 correct formatted results are selected for the results calculation. From more than 55 companies people responded to the survey.

3.2.2- Role/Designation

Different roles and designation have filled the survey including software developers, project managers, team leads, senior and junior level software engineer, quality assurance engineers working on different nature of projects from different organizations. It was an open question and people replied to this with their current role as they are assigned in their organization. As this was not compulsion to fill the questionnaire so we have got scattered results. On average IT professionals from all fields played their part in filling the survey. Mostly the responses are from software development and software quality assurance and technical support people with core involvement in software development life cycle.

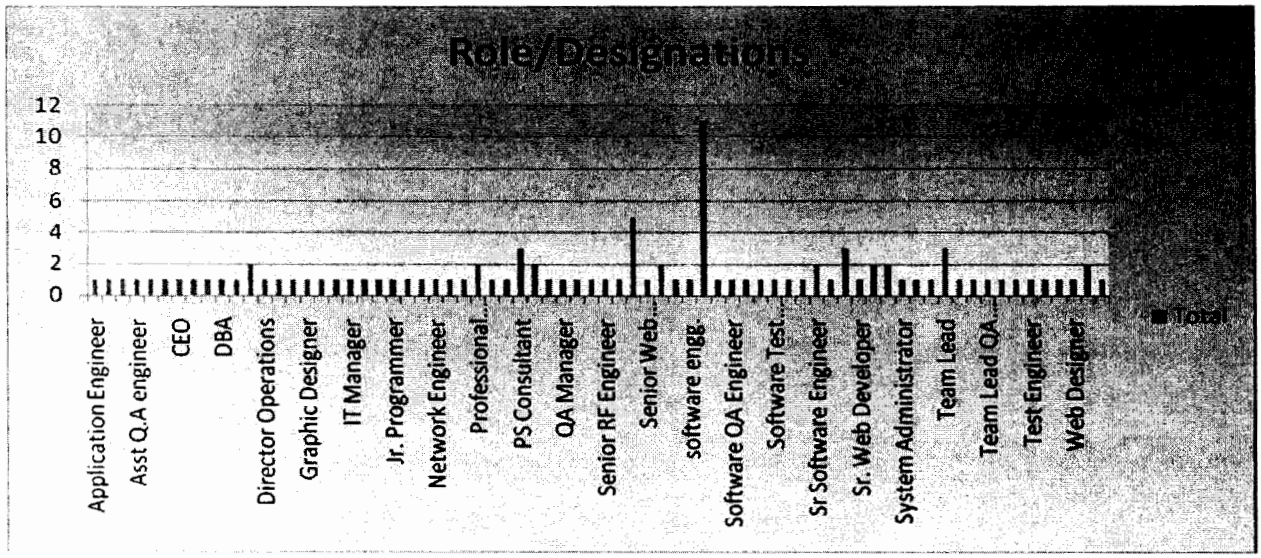


Fig 1. Role/Designation of

3.2.3- Organization Size

The target market of the survey is Pakistan software industry and fortunately we have a great diversity in it ranging from different project size to organization size. We have very large companies with more than 5000 employees and small organization having teams of 15-10 people.

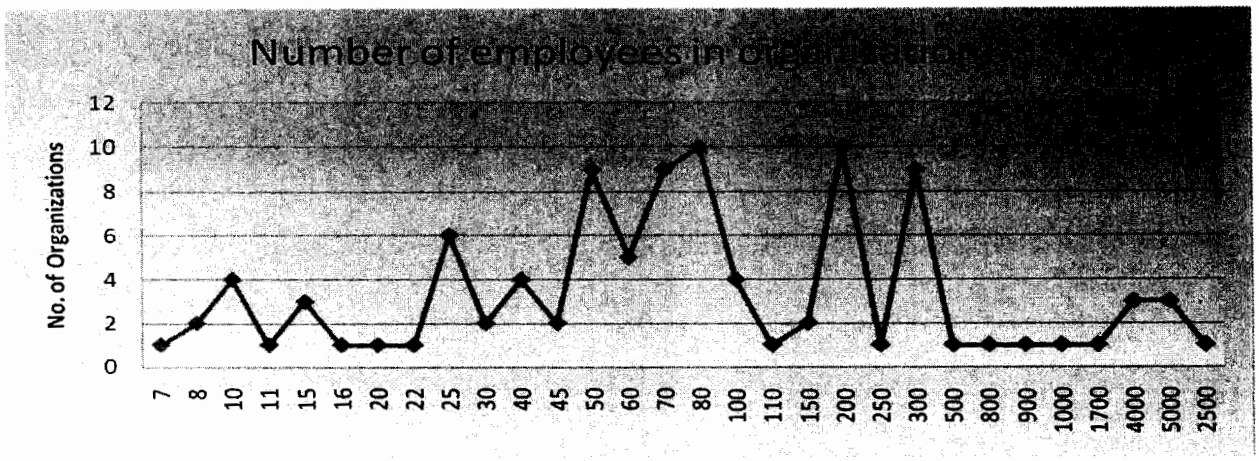


Fig 2. Organization size

3.2.4- Software Model/Technique Used

This question was asked to identify if there is a software development or specific software development technique is used in the organization for the software projects. Majority of the organization participated in the survey i.e. 57% have replied with yes that they are using some software development model or technique.

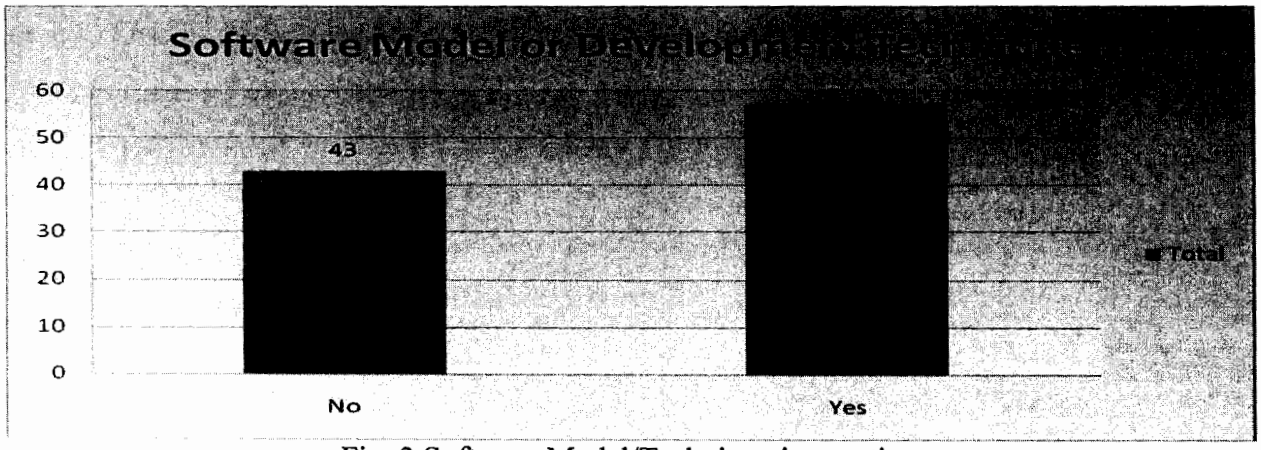


Fig. 3 Software Model/Technique in practice

3.2.5- Software Product line

Reusability is more important for the products with same feature or a software product line. The purpose of this question was to inquire how many organizations are working on software product line so we can get the need of reusability or reusability model needs from the market. Fortunately more than 70% responses are positive. Majority of the organization of Pakistan software industry are working on software product line.

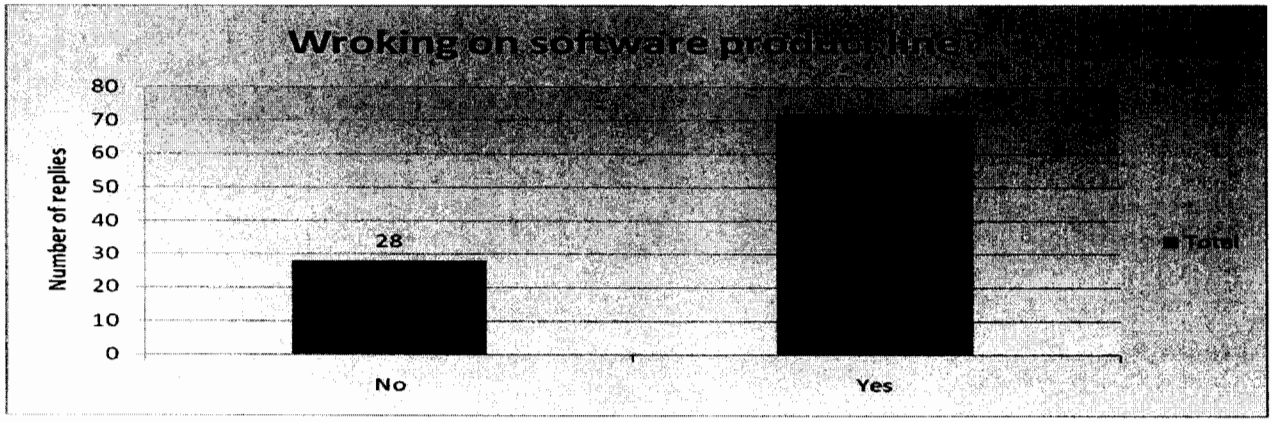


Fig. 4 Working on Software Product Line

3.2.6- Software Module/Components reused

This question was regarding the reusability in practice for software modules as a whole or software components in software industry of Pakistan. 23% of the respondent strongly agreed with the reusability of software components. 59% replied with agree, 15% neutral and only 3% replied with disagree. The result s showed that mostly the software components and modules are reused in market whether a software model is followed or not. So already knowingly or unknowingly reusability is in practice.

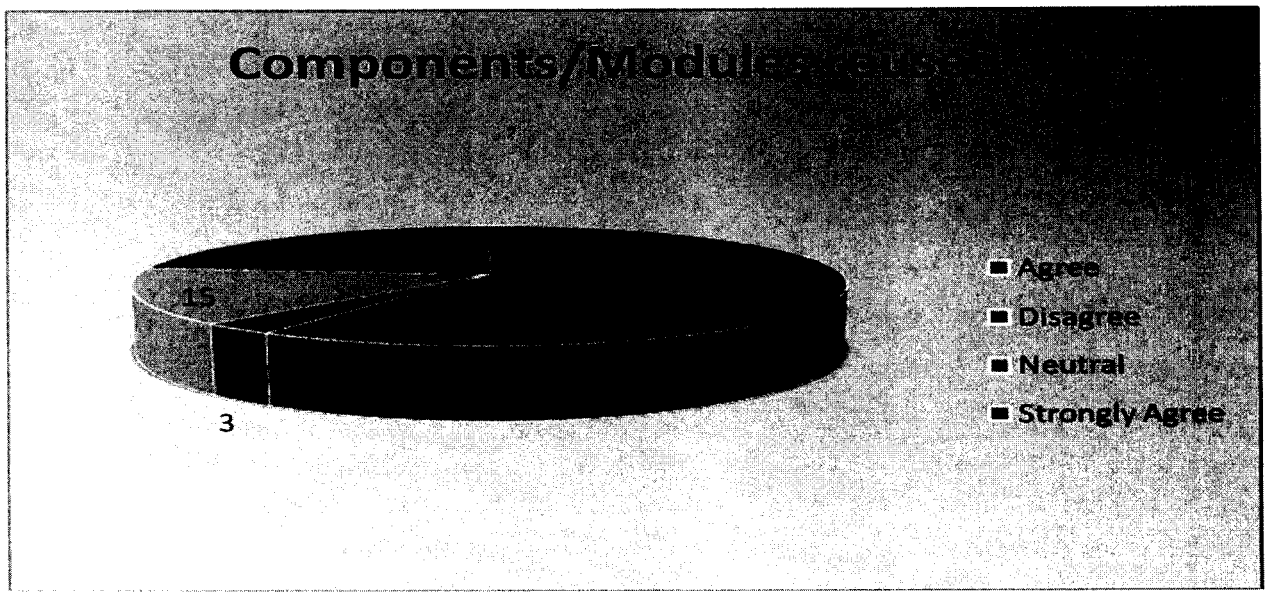


Fig. 5 Module/Components reused

3.2.7- Flexibility:

We have selected a model that proposes that reusability is dependent on different attributes. Flexibility is one of them. We have come to know so far from the results that reusability is in practice in some form. Now we want to know that is it really dependent on the attributes mentioned in the model or not. Users have been asked about these attributes and in the question. This part is about the flexibility hat how flexible are the software solution they provide for different clients. Majority of the replies are positive that their solution is that flexible to be offered to different clients with or without any modification.

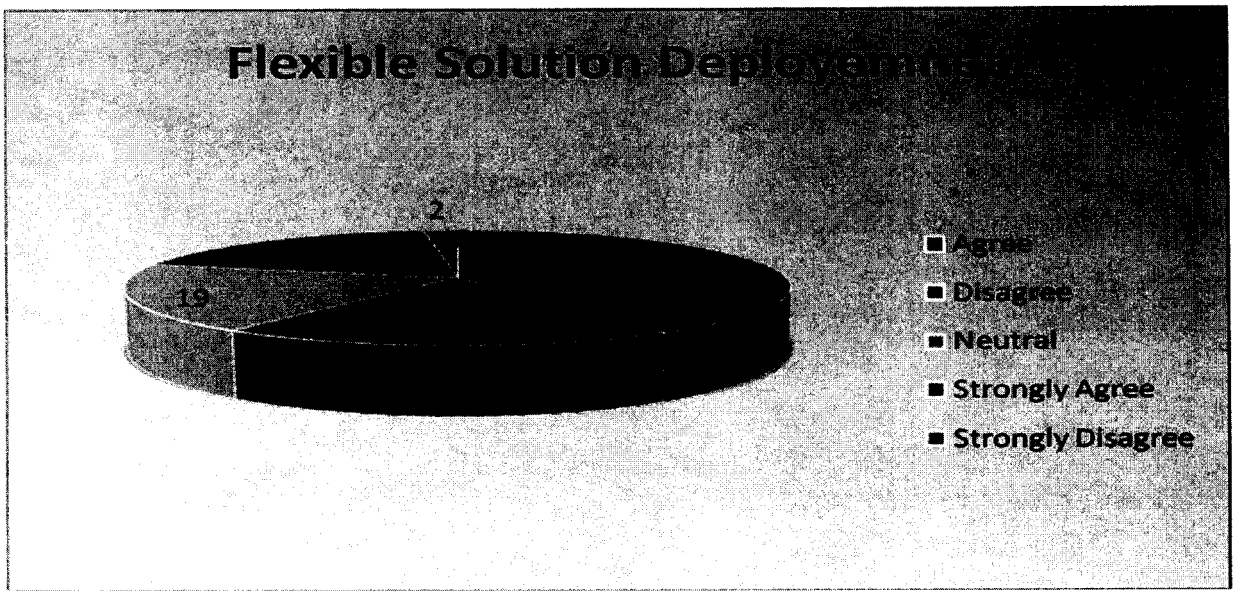


Fig 6. Flexibility

3.2.8- Maintainability

Our selected model assumes that reusability is dependent on the level of maintainability of the software product. We have asked the respondents about the practice of maintainability keeping the view in mind that reusability practice are used. 47% replied with agree that there is some mechanism for maintainability of software release. 40% replied with strongly agree, 9% with neutral, 2% each of disagree and strongly disagree.

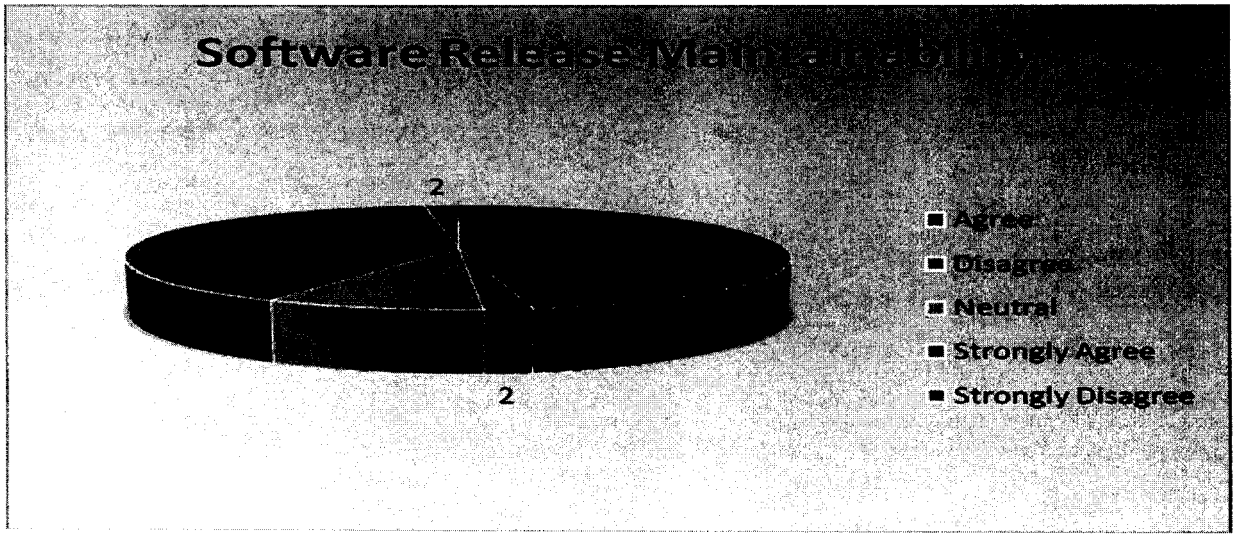


Fig 7. Maintainability

3.2.9- Portability

As portability is another attribute on with reusability is dependent according to the selected model. We have asked using our survey from the market if is portability is important for software reusability. 49% of the responses replied with agree and 10% with strongly agree. 29% shown neutral response, 1% strongly disagrees and 11% disagree.

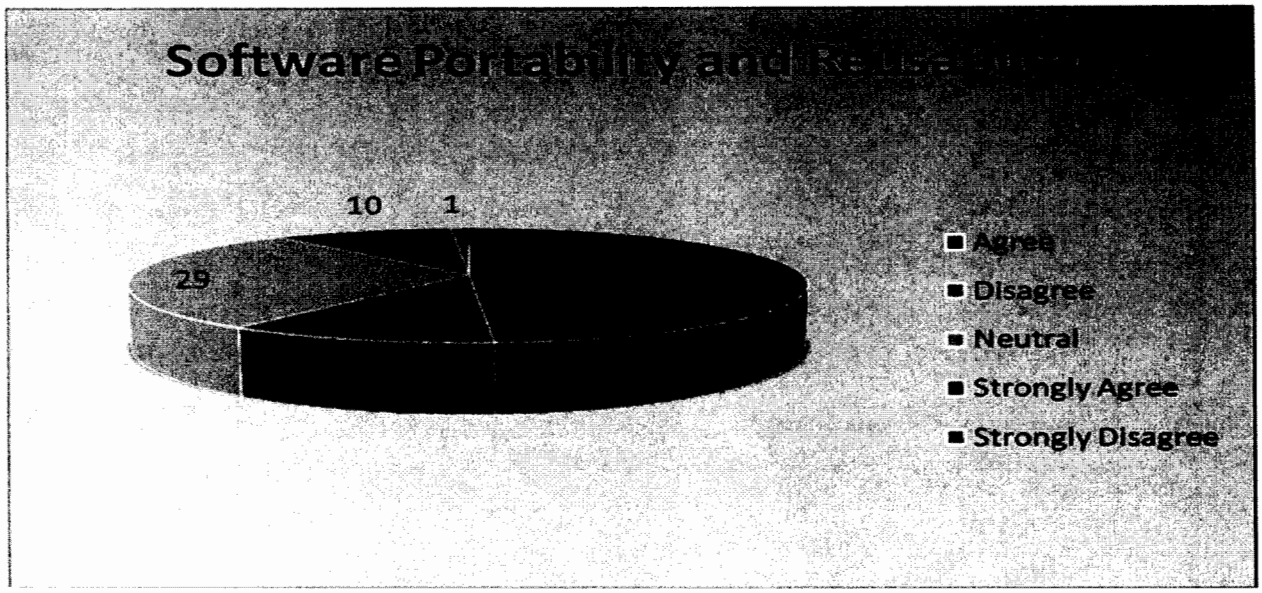


Fig 8. Portability & Reusability

3.2.10- Understabibility

Users were asked about the relationship between software reusability and software understandability. Reusability can help to increase more understandability. Users have responded to this question in positive and 62% are in agreed. 25% replied with neutral and 7% with disagree and 6% with strongly agree.

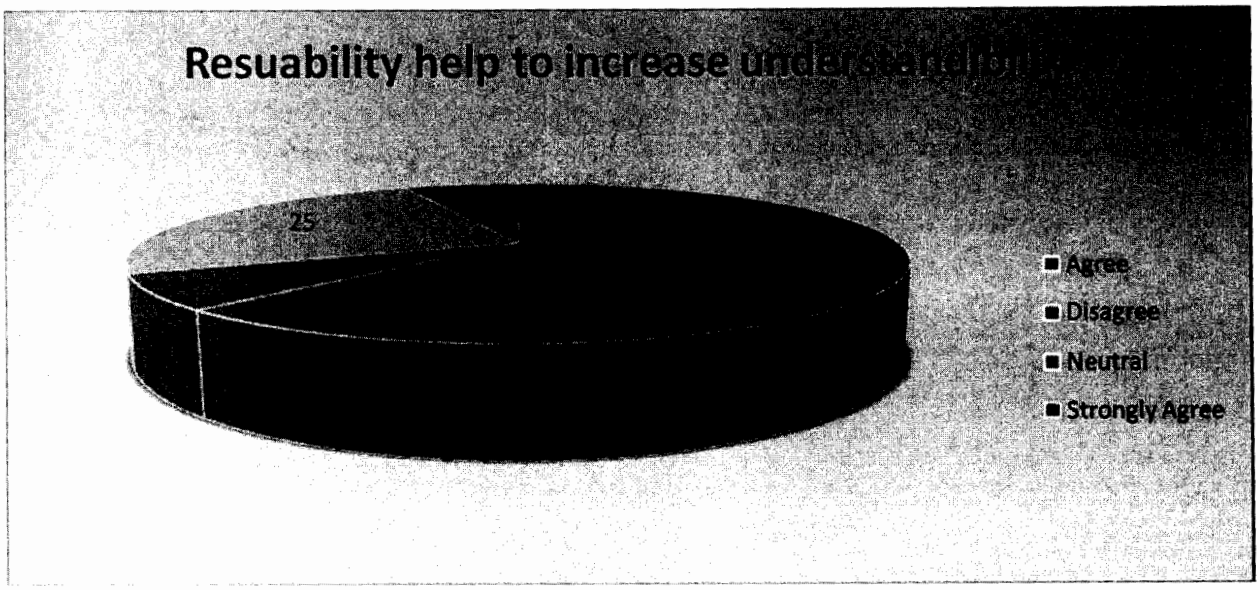


Fig. 9 Reusability and Understandability

3.2.11- Scope Coverage

In survey we asked about how reusability is in practice for scope coverage in different project. Our selected model proposes that reusability is dependent on scope coverage as well. So the market is asked about the reusability of business logic in multiple projects/products.61% of the replies are in agree, 13% strongly agree, 15% neutral, 7% disagree and 4% strongly disagree.

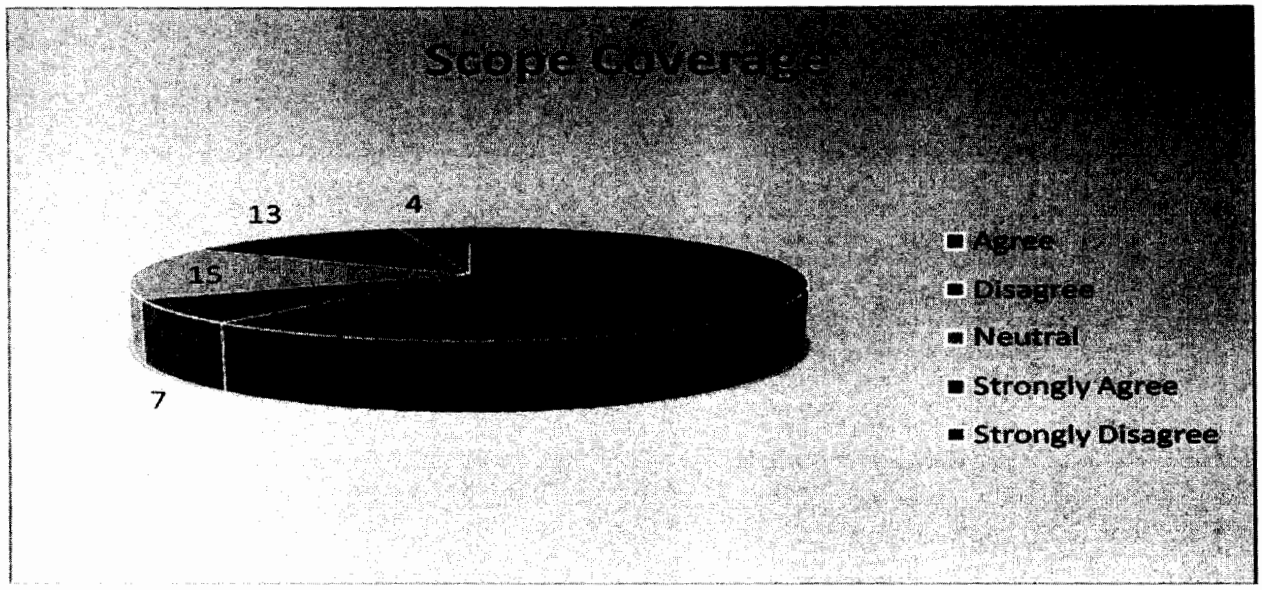


Fig. 10 Scope Coverage

3.2.12- Modular Software Development Approach

Users were asked about if the development in their organization for software products is modular or not. So from the result of survey and keeping in mind the model of reusability with respect to AOSD we need software development to be modular. So we can identify reusability as separate concern. Fortunately we have got positive response for this. In most of the organization the approach used is modular. 62% replied with agree and 21% replied with strongly agree. Only 4% replied with disagree and 13% of the responses were neutral. So the modular approach is mostly used in software industry.

Fig. 11 Modular Development

3.2.13- Separation of concerns

Aspect oriented software development is about separation of cross cutting concerns. Mostly industry and professional are unaware of the separation of the concerns. So it was presented in simpler form and was asked that is there separation of concerns. We have got 47% neutral response for this and 39% agree.6% replied with strongly agree and 7% with disagree. The results show that users are not sure about the separation of concerns.

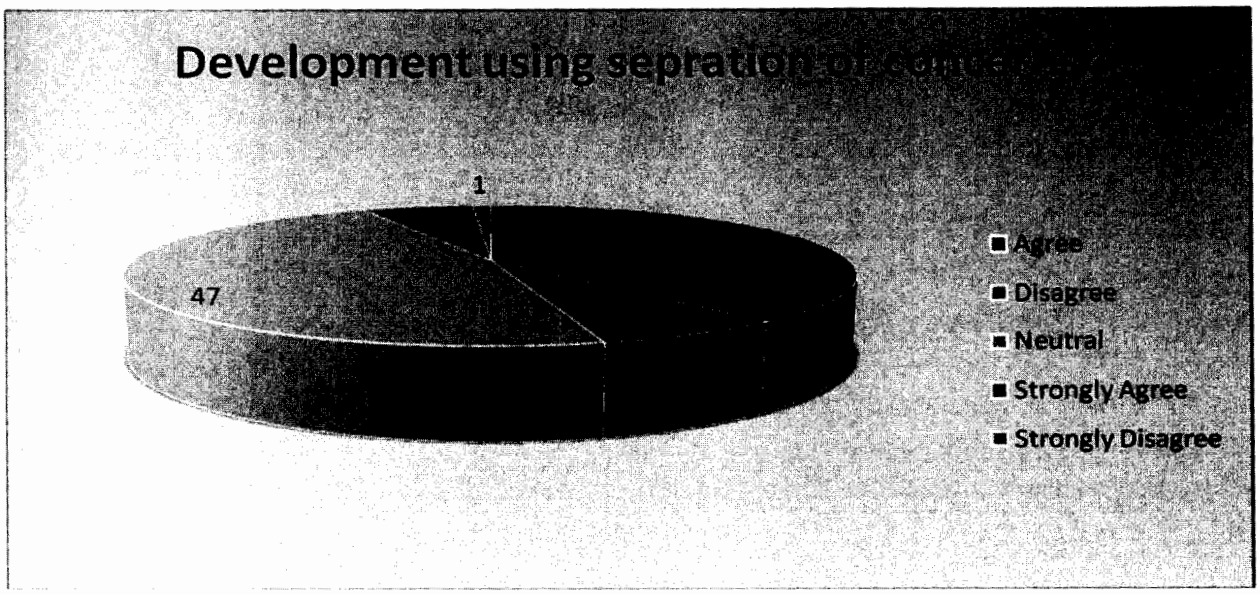


Fig. 12 Sepration of concerns

3.2.14- Importance of reusability

This section covers more than one question asked form the people about the importance of reusability. People are asked in different ways about what is the significance of reusability and is there and need of a model for software development focusing on separation of concerns and dealing reusability as separate concern. Overview and results of the question asked about importance of reusability is as follow.

Users are asked is reusability is the key of software development. 55% replied with agree and 31% with strongly agree. 8% responses were neutral and 5% disagreed and only 1% strongly disagreed. The result of this question showed that people know about the importance of reusability and its significance.

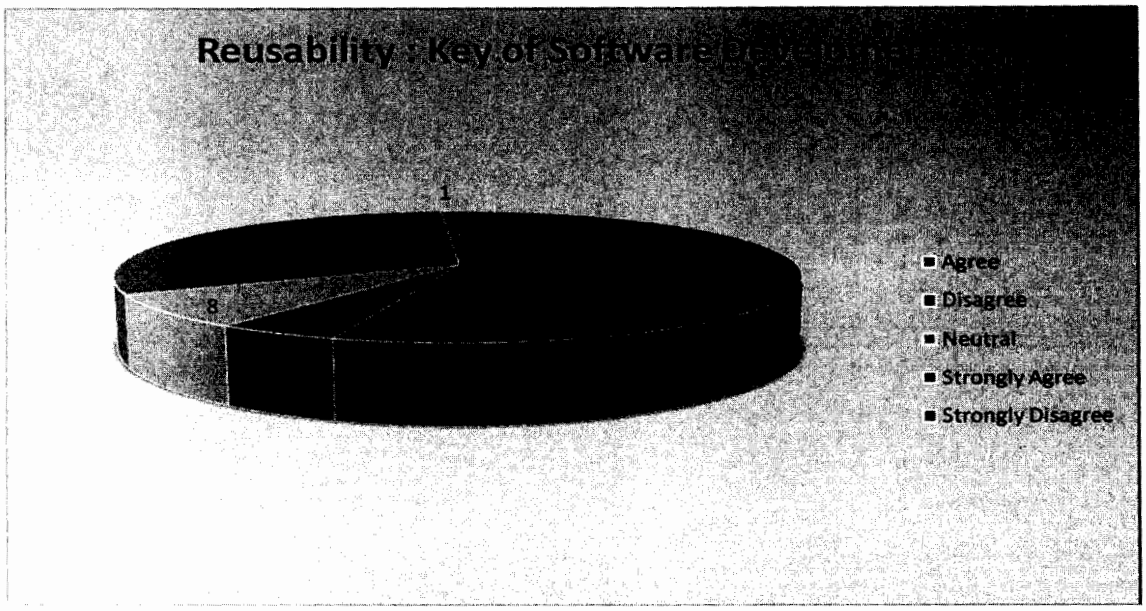


Fig. 13 Reusability as key

Users were asked if reusability is in practice then will it help to reduce the time of development. It is obvious if components and modules are reused then it will save the time to reinvent the wheel. 50% responses strongly agreed and 43% agreed. 7% showed neutral behavior and none replied with disagreed with this question.

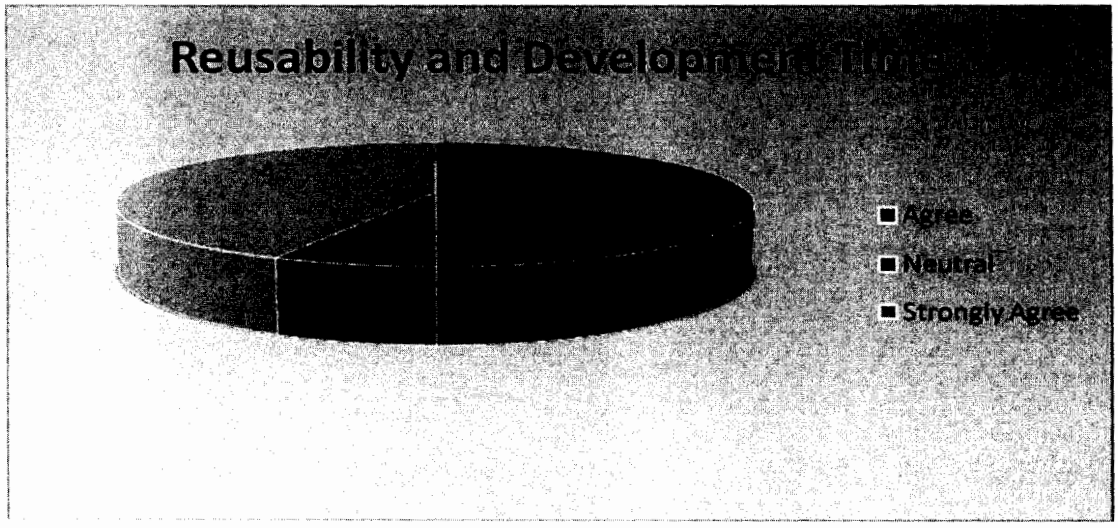


Fig. 14 Reusability and Development Time

Chapter: 4. Survey Results and Fuzzy Logic

4- Survey Result Analysis

The market survey is done keeping view the model of reusability that is based on different attributes with respect to Aspect Oriented Software Development. We have taken the results based on those attributes and converted them into numeric form to formulate an analysis of the results to implement the next phase of the research i.e. applying the Fuzzy Logic on them to verify the expected output. We have selected flexibility, portability, understandability, scope coverage, variability and modifiability and given them a range from 1-5 to apply Fuzzy Logic. The scale is as follow:

1=Strongly Agree

2=Agree

3=Neutral

4=Disagree

5=Strongly Disagree

Moreover we have asked the market about the importance and need of reusability to get a clear picture about the need of a model that addresses reusability with respect to Aspect Oriented Software Development. We have concluded that reusability practices are used and there are some sorts of techniques, methodologies and model used in the market to address reusability. Our study will help in modeling those techniques to address reusability as separate concern. We have modified our survey results to implement the next step of our study.

4.1 Fuzzification of results

4.1.1 What is Fuzzy Logic?

Fuzzy logic is a logical system, which is an extension of multivalve logic and broadly speaking it is synonymous with fuzzy sets, a theory related to classes of objects with unsharp boundaries in which membership is a matter of degree. [25].

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth- truth values between "completely true" and "completely false". As its name suggests, it is the logic underlying modes of reasoning which are approximate rather than exact. The importance of fuzzy logic derives from the fact that most modes of human reasoning and especially common sense reasoning are approximate in nature [26].

4.1.2 Why Fuzzy Logic?

Fuzzy logic is an easy concept to understand. We have to compare reusability with respect to AOSD and AOSD is not in practice in the industry as we have our survey results. We have to base our result on the base of human communication so Fuzzy Logic is the best option closest to human communication. Fuzzy Logic somehow is based on natural language. From market survey it is clear that AOSD is not in practice in the industry.

Binary Logic is represented as Yes or No. It is easy to make decision on the basis of 1 and 0 because the options are straight.

Fuzzy Logic is used for reasoning when we are not having straight results. We do not have results in the form of completely true or completely false.

The results of market survey are that cannot be used precisely to measure the reusability. We have used Fuzzy Logic because the attributes on which the Reusability is based are not in the form of Yes or No. They could be very low, low, medium, high and very high in importance.

Results of my survey are in the form of range from 1-5 not in the form of 1 and 0. As my survey results vary on a scale. So I have fuzzified the result to reach the conclusion the degree of reusability achieved by changing attribute range.

I have developed the fuzzy logic application on the basis of ling variables based upon member functions. Depending upon the low, medium and high degrees of my attributes altogether on which reusability is dependent, the results of degree of reusability varies directly.

As in the case above the importance of the attributes on which Fuzzy is based are not straight and ambiguous in nature. So the Fuzzy logic implementation is feasible to attain the approximate results.

If I have to apply my survey results to the binary logic then I have to alter my results to list them as true or false. [27][25].

4.1.3 Fuzzy Logic Techniques

We have used Mamdani method thate covers Fuzzy Logic for fuzzification of our survey result on the basis of rules of our selected model of software reusability.

Mamdami algorithm can be described in our context as follow:

Start

Define Fuzzy Variables

Determine fuzzy variables, reusability factor, literature Priority and survey [users] Priority

Establish fuzzy sets for these variables

Fuzzify each value in fuzzy sets using membership function

Generate knowledge Base using fuzzy rules

Build the system

Execute the system

Give input variable values

Get rule strength

Combine rule strength with output membership function

Find consequence of rules

Generate output

Combine consequences of variant rules

Generate output distribution by conflict resolution process

Defuzzify

3. *Finish*

4.1.4 Fuzzification of results:

From the survey result we have selected different range of values for the factor on which reusability is dependent. Those factors are variability, understandability, scope coverage, maintainability, portability and flexibility. The overall purpose of the survey was to understand the practices of reusability in the market and impact of these attribute son reusability. We have the results of the survey are displayed in graphical form below.

Variability:

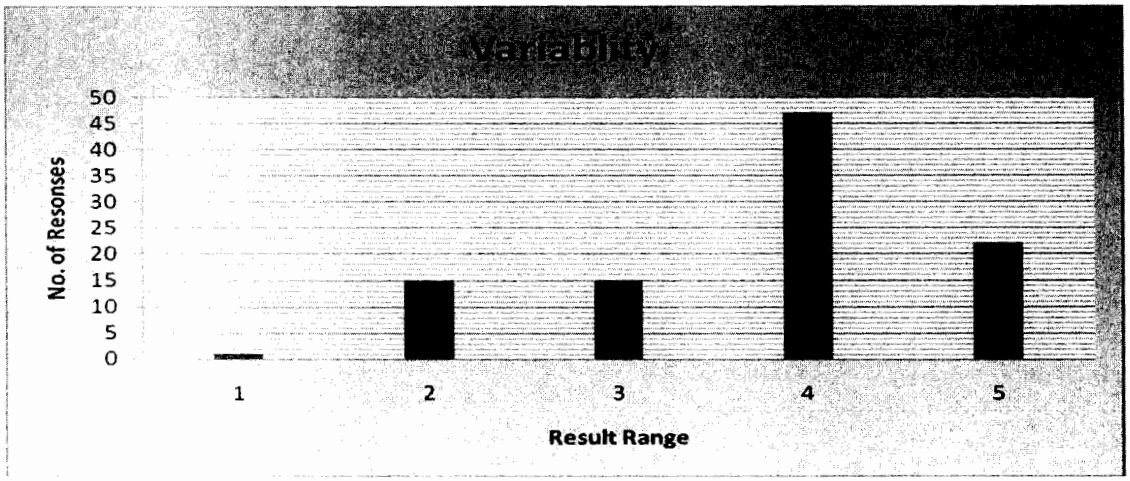


Fig. 14 (Scale: 1= Strongly Disagree 2=Disagree 3=Neutral 4=Agree 5=Strongly Agree)

Portability:

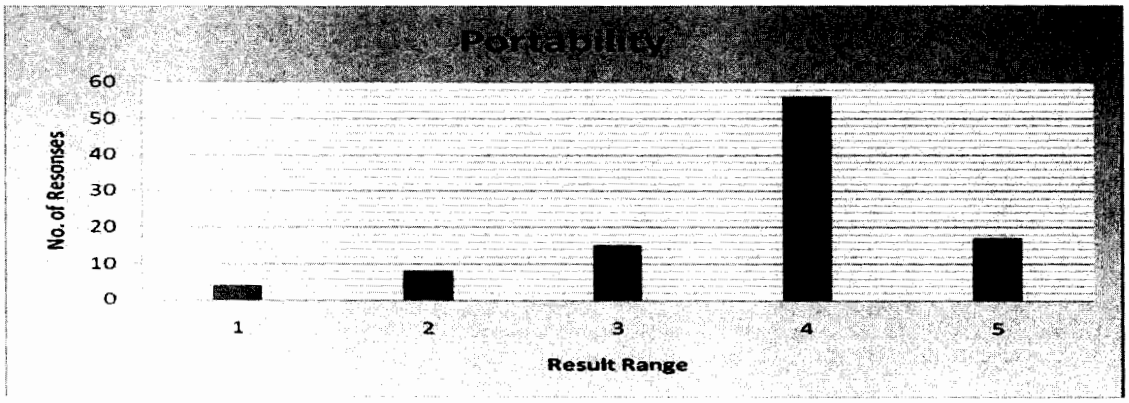


Fig. 15 (Scale: 1= Strongly Disagree 2=Disagree 3=Neutral 4=Agree 5=Strongly Agree)

Maintainability:

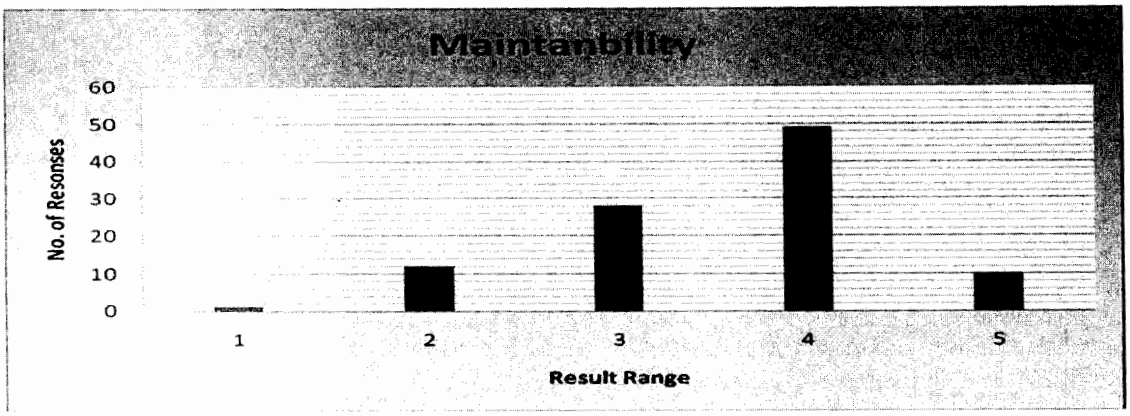


Fig. 16 (Scale: 1= Strongly Disagree 2=Disagree 3=Neutral 4=Agree 5=Strongly Agree)

Understandability:

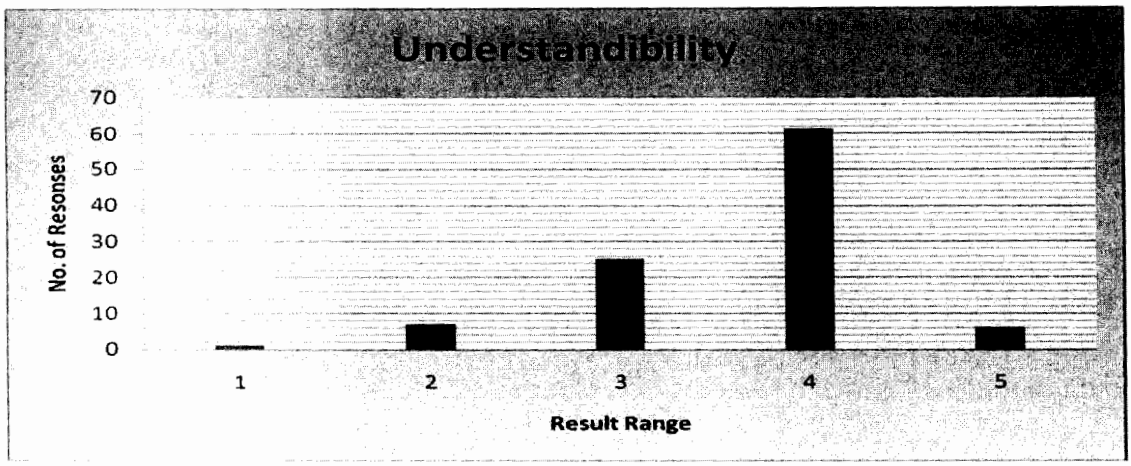


Fig. 17 (Scale: 1= Strongly Disagree 2=Disagree 3=Neutral 4=Agree 5=Strongly Agree)

Flexibility:

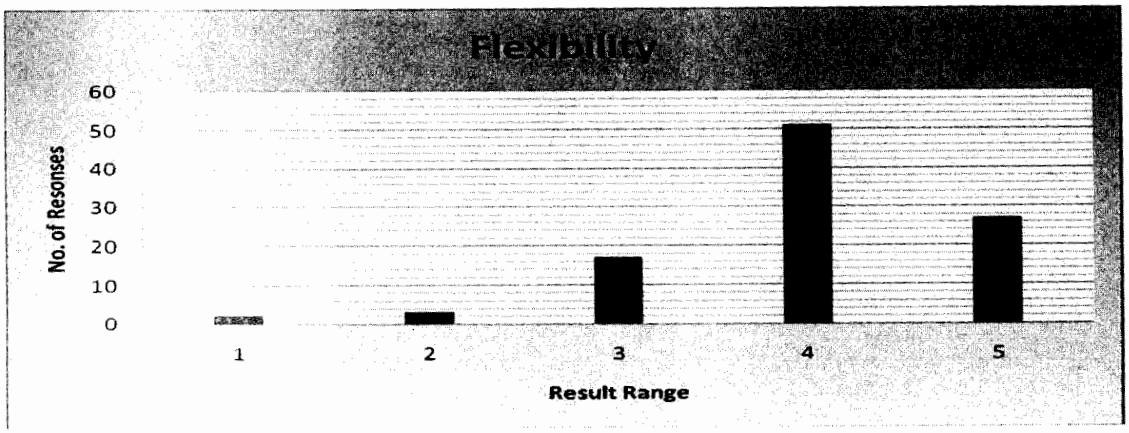


Fig. 18 (Scale: 1= Strongly Disagree 2=Disagree 3=Neutral 4=Agree 5=Strongly Agree)

Scope Converge:

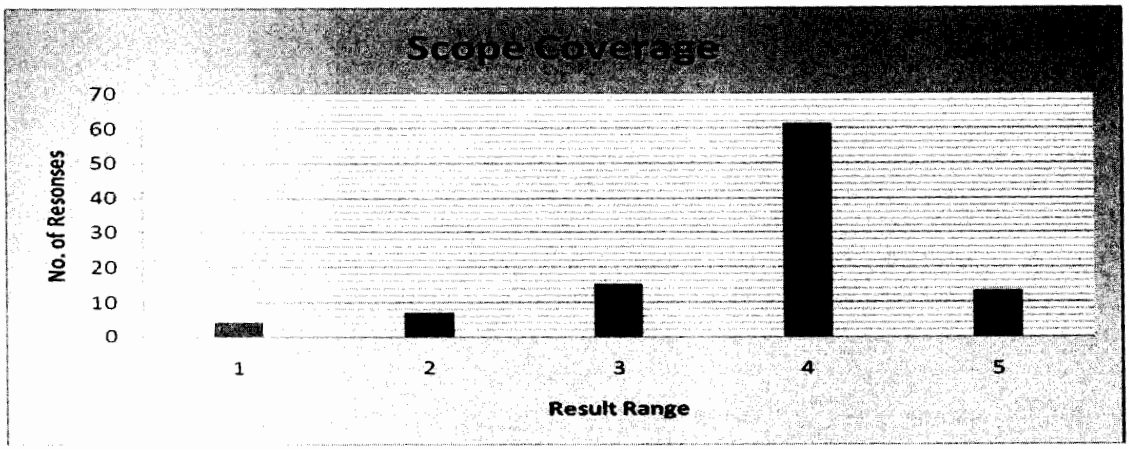


Fig. 19 (Scale: 1= Strongly Disagree 2=Disagree 3=Neutral 4=Agree 5=Strongly Agree)

On the basis of above results we have made our member functions to apply fuzzy logic. We have developed an application 'Fuzzyfier' on the basis of Mamdami Approach to put our result and converts into output in the form of reusability percentage.

4.1.5 Fuzzy Variables:

Fuzzy logic is different from normal sets calculation where we have 1 and 0 situation. In Fuzzy logic we have range from lower to medium to higher. Fuzzy sets can be defined for either finite or infinite domains. On the basis of above mentioned algorithm we have derived our fuzzy variables from the model and the values are derived from the survey results filled by the users from software industry in Pakistan. These variables basically are the attributes on which reusability is dependent. So our fuzzy variable on which the reusability is dependent are the same on which our model is based.

$$VL_{rv} = [a_{VL_{rv}}, b_{vl_{rv}}, c_{vl_{rv}}, d_{vl_{rv}}]$$

$$L_{rv} = [a_{L_{rv}}, b_{l_{rv}}, c_{l_{rv}}, d_{l_{rv}}]$$

$$M_{rv} = [a_{M_{rv}}, b_{M_{rv}}, c_{M_{rv}}, d_{M_{rv}}]$$

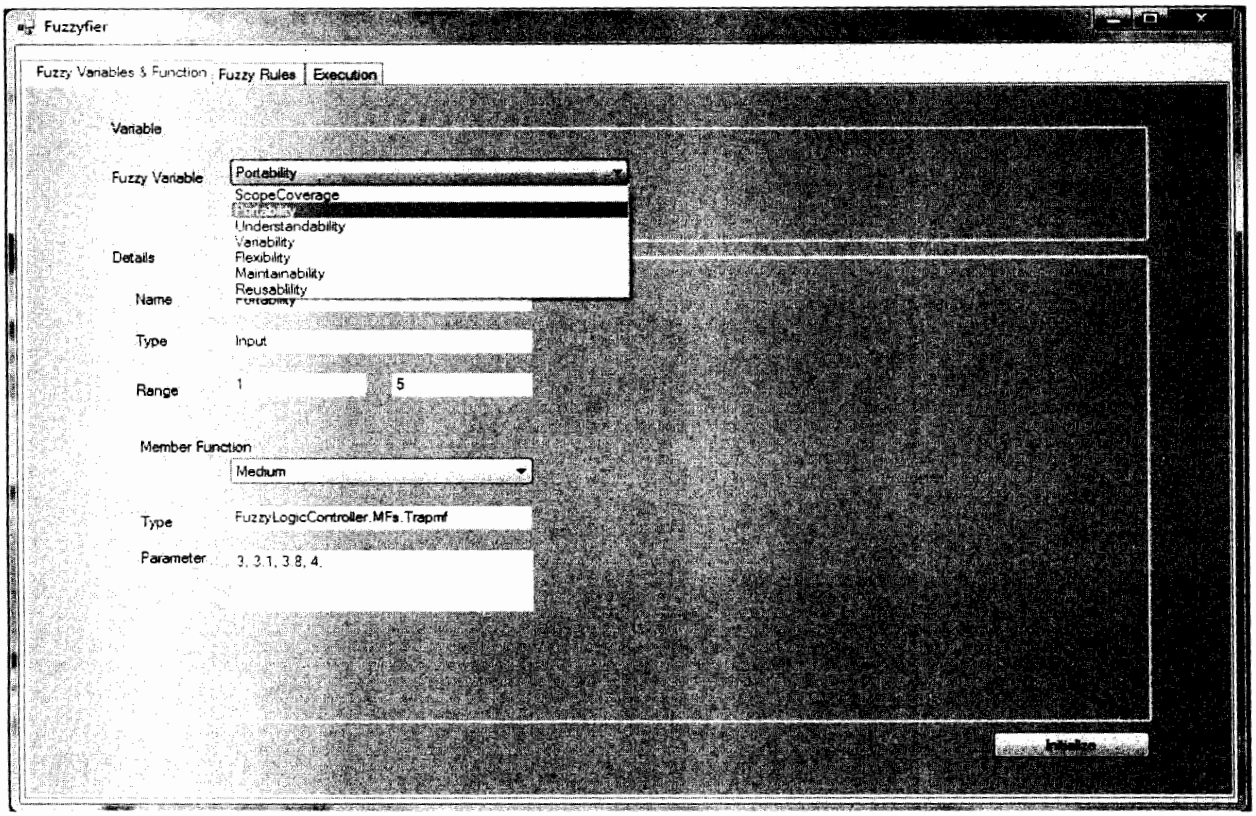
$$H_{rv} = [a_{H_{rv}}, b_{H_{rv}}, c_{H_{rv}}, d_{H_{rv}}]$$

$$VH_{rv} = [a_{VH_{rv}}, b_{VH_{rv}}, c_{VH_{rv}}, dV_{H_{rv}}]$$

We have defined our Fuzzy Variables in the application as follow:

```
public LingVariable getVar(String name)

        {

            if (name.Equals("ScopeCoverage")) { return X1; }

            else if (name.Equals("Portability")) { return X2; }

            else if (name.Equals("Understandability")) { return X3; }

            else if (name.Equals("Variability")) { return X4; }

            else if (name.Equals("Flexibility")) { return X5; }

            else if (name.Equals("Maintainability")) { return X6; }

            else if (name.Equals("Reusablility")) { return Y1; }
```

```
        return null;
    }
```



4.1.6 Member Function:

Member functions in fuzzy logic are basically the characteristics function to describe the fuzzy variables or fuzzy sets. The objective of member functions is to define the degree of the fuzzy variables. There are some rules to define member functions. There is no concept of negative member function they always start from 0. We have to decide a benchmark for making decision about the degree of the fuzzy variable.

The fuzzy member function are defined in application as follow:

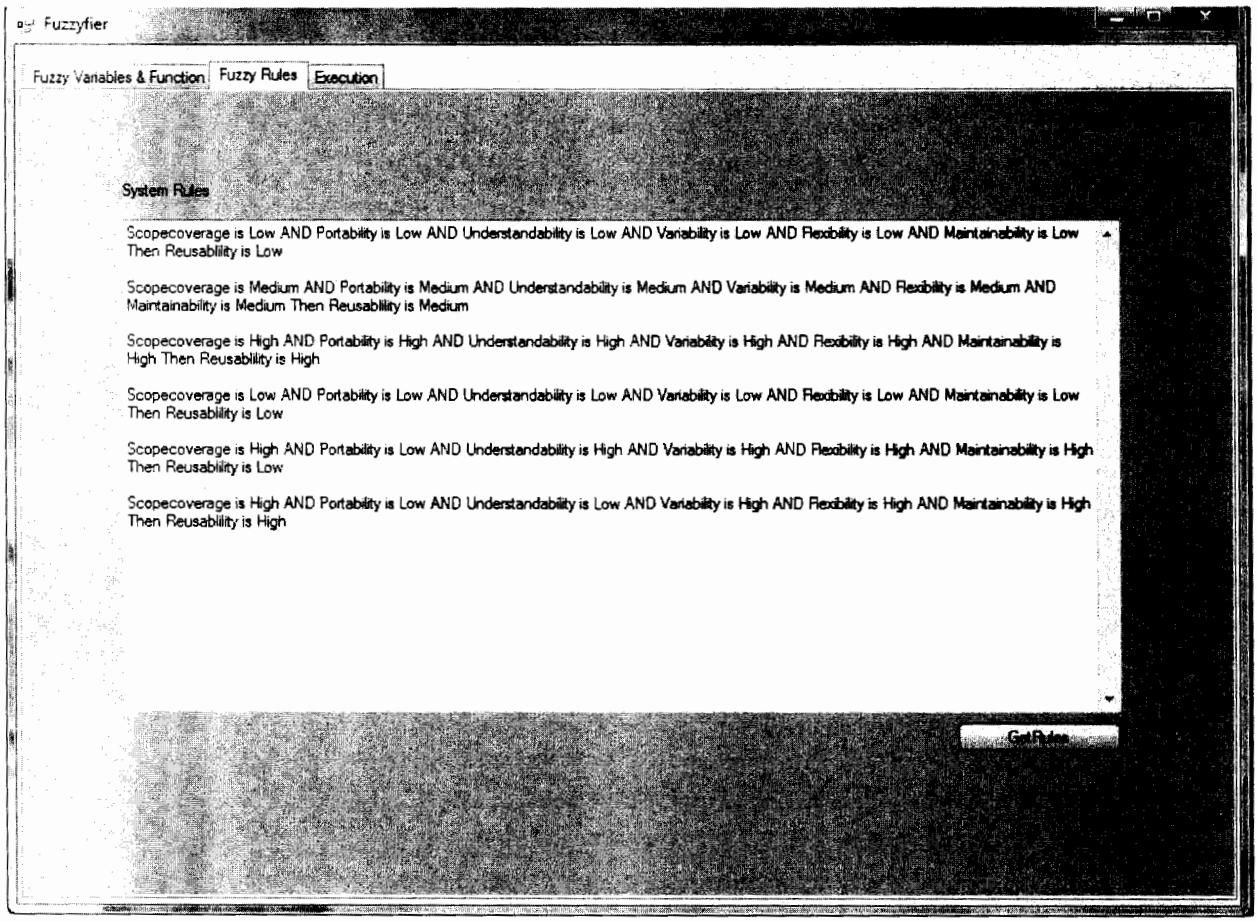
```
public String makeRules()
        {
```

```
String rulesStr = "";

rulesStr += "Scopecoverage is Low AND Portability is Low AND
Understandability is Low AND Variability is Low AND Flexibility is Low AND
Maintainability is Low Then Reusablility is Low" +
System.Environment.NewLine;

rulesStr += "" + System.Environment.NewLine;

rule1in.AddRange(new RuleItem[6] {

    new RuleItem("Scopecoverage", "Low"),

    new RuleItem("Portability", "Low"),

    new RuleItem("Understandability", "Low"),

    new RuleItem("Variability", "Low"),

    new RuleItem("Flexibility", "Low"),

    new RuleItem("Maintainability", "Low") });

rule1out.AddRange(new RuleItem[1] { new RuleItem("Reusablility",
"Low") });


rulesStr += "Scopecoverage is Medium AND Portability is Medium AND
Understandability is Medium AND Variability is Medium AND Flexibility is
Medium AND Maintainability is Medium Then Reusablility is Medium" +
System.Environment.NewLine;

rulesStr += "" + System.Environment.NewLine;

rule2in.AddRange(new RuleItem[6] {

    new RuleItem("Scopecoverage", "Medium"),

    new RuleItem("Portability", "Medium"),

    new RuleItem("Understandability", "Medium"),

    new RuleItem("Variability", "Medium"),
```

```
                new RuleItem("Flexibility", "Medium"),
                new RuleItem("Maintainability", "Medium") });
            rule2out.AddRange(new RuleItem[1] { new RuleItem("Reusablility",
"Medium") });
```

The output of the above rules is as follow in the application.

- Scope coverage is Low AND Portability is Low AND Understandability is Low AND Variability is Low AND Flexibility is Low AND Maintainability is Low Then Reusability is Low
- Scope coverage is Medium AND Portability is Medium AND Understandability is Medium AND Variability is Medium AND Flexibility is Medium AND Maintainability is Medium Then Reusability is Medium
- Scope coverage is High AND Portability is High AND Understandability is High AND Variability is High AND Flexibility is High AND Maintainability is High Then Reusability is High
- Scope coverage is Low AND Portability is Low AND Understandability is Low AND Variability is Low AND Flexibility is Low AND Maintainability is Low Then Reusability is Low
- Scope coverage is High AND Portability is Low AND Understandability is High AND Variability is High AND Flexibility is High AND Maintainability is High Then Reusability is Low
- Scope coverage is High AND Portability is Low AND Understandability is Low AND Variability is High AND Flexibility is High AND Maintainability is High Then Reusability is High



4.1.7 Member Function Range

On the basis of the results of our survey we have set the range of our member function from 1 to 5 on the basis of result options. On the basis of this range the values assigned to the variable to fuzzify the results are as follow:

Low: 0, 1, 2, 3

Medium: 3, 3.1, 3.8, 4

High: 3.7, 4, 4.5, 5.9

4.1.8 Linguistic Variables

Lotfi Zadeh introduced the concept of linguistic variable or fuzzy variable in 1973 in order to facilitate computation with words instead of numbers [40]. These variables have values that are words or sentences from natural language. For example, sensory inputs are linguistic variables,

or nouns in a natural language e.g. temperature, pressure, displacement, etc. The purpose of these variables is to allow the translation of natural language into logical or numerical statements for approximate reasoning.

Linguistic variables can be divided into different categories:

- Quantification variables, e.g. all, most, many, none, etc.
- Usuality variables, e.g. sometimes, frequently, always, seldom, etc.
- Likelihood variables, e.g. possible, likely, certain, etc.

In natural language, nouns are frequently combined with adjectives for quantifications of these nouns. By using adjectives along with nouns, we can differentiate between two elements in a qualitative manner. For example, the phrase very high can be used to indicate an element which is higher than high. The term hedge is used to refer to these adjectives. Hedges work as modifier of fuzzy values. These are implemented through subjective definitions of mathematical functions, to transform membership values in a systematic manner.

```
public void InitFuzzyVars()

        {

            X1 = new LingVariable("Scopecoverage", VarType.Input);

            X1.setRange(1, 5);

            X1.addMF(new Trapmf("Low", 0, 1, 2, 3));

            X1.addMF(new Trapmf("Medium", 3, 3.1, 3.8, 4));

            X1.addMF(new Trapmf("High", 3.7, 4, 4.5, 5.9));


            X2 = new LingVariable("Portability", VarType.Input);

            X2.setRange(1, 5);
```

```
X2.addMF(new Trapmf("Low", 0, 1, 2, 3));

X2.addMF(new Trapmf("Medium", 3, 3.1, 3.8, 4));

X2.addMF(new Trapmf("High", 3.7, 4, 4.5, 5.9));


X3 = new LingVariable("Understandability", VarType.Input);

X3.setRange(1, 5);

X3.addMF(new Trapmf("Low", 0, 1, 2, 3));

X3.addMF(new Trapmf("Medium", 3, 3.1, 3.8, 4));

X3.addMF(new Trapmf("High", 3.7, 4, 4.5, 5.9));


X4 = new LingVariable("Variability", VarType.Input);

X4.setRange(1, 5);

X4.addMF(new Trapmf("Low", 0, 1, 2, 3));

X4.addMF(new Trapmf("Medium", 3, 3.1, 3.8, 4));

X4.addMF(new Trapmf("High", 3.7, 4, 4.5, 5.9));


X5 = new LingVariable("Flexibility", VarType.Input);

X5.setRange(1, 5);

X5.addMF(new Trapmf("Low", 0, 1, 2, 3));

X5.addMF(new Trapmf("Medium", 3, 3.1, 3.8, 4));

X5.addMF(new Trapmf("High", 3.7, 4, 4.5, 5.9));
```

```
        X6 = new LingVariable("Maintainability", VarType.Input);
        X6.setRange(1, 5);
        X6.addMF(new Trapmf("Low", 0, 1, 2, 3));
        X6.addMF(new Trapmf("Medium", 3, 3.1, 3.8, 4));
        X6.addMF(new Trapmf("High", 3.7, 4, 4.5, 5.9));

        Y1 = new LingVariable("Reusablility", VarType.Output);
        Y1.setRange(0, 100);
        Y1.addMF(new Trapmf("Low", 0, 20, 30, 40));
        Y1.addMF(new Trapmf("Medium", 41, 50, 60, 70));
        Y1.addMF(new Trapmf("High", 71, 85, 100, 103));
    }
```

4.1.9 Fuzzification Process

Fuzzification is the process of evaluating the crisp input against the membership functions of the lingstic variable Scope Coverage membershipfunction is an abstract class, with an abstract method "getOutput", any object inheriting this class needs to define their getOutput(double) function, such as in Trimf and Trapmf. Based upon this feature, more membershipfunctions can be implemented to extend the fuzzy controller capabilities.

Fuzzification method then returns a list of fuzzy numbers, each fuzzy number consists of a function name and firing strength from getOutput(double).

Example: fuzzynumber(0,2.9,"Low") FuzzyNumber(4,5.9,"High")

The list of fuzzy numbers is then added to a fuzzy Set with list of Fuzzy numbers and linguistic Variable name for further processing.

The rules are pallied on the ranges of variables and results is formed on the basis of the variable ranges.

4.1.10 Rules Evaluation

On the basis of strength of the variables, the strength of the output is measured

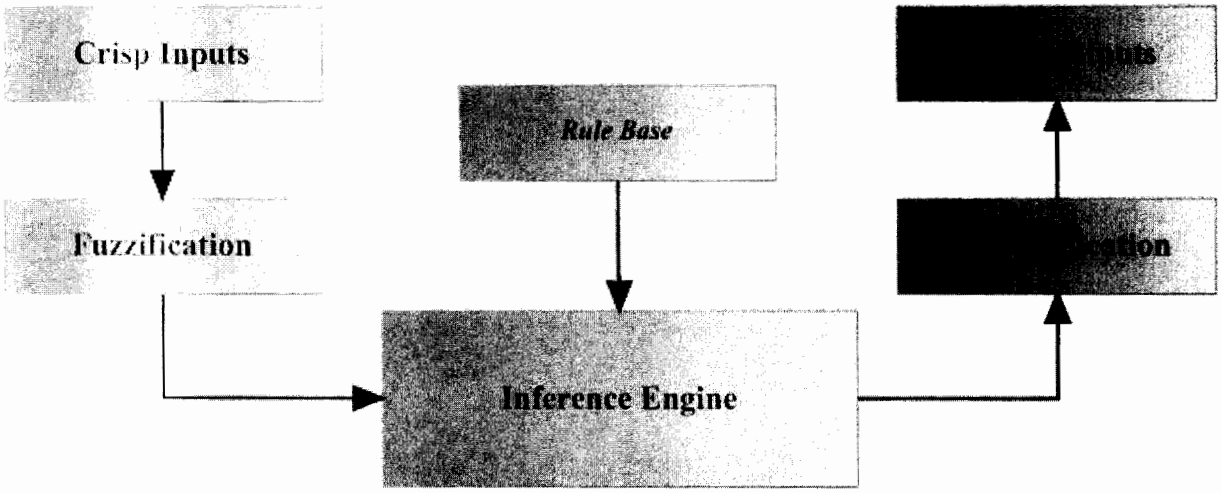
For Example in our case

Rule: Scope coverage is Low AND Portability is Low AND Understandability is Low AND Variability is Low AND Flexibility is Low AND Maintainability is Low Then Reusability is Low

4.1.11 Defuzzification of results

When the rules evaluate the input and generate and output, it is again converted into some crisp value to get an absolute value.

The whole process can be represented in figure below.



We get the defuzzified value of the consequent using the following equation:

$$y_i = \frac{\sum_{j=1}^{p} y_j \mu_A(y_j)}{\sum_{j=1}^{p} \mu_A(y_j)}$$

Results Set: On the basis of our application few results are as follow in the table

Range	Variable						
	Variability						
1	VH	VH	VH	VH	VH	VH	VH
2	H	H	H	H	H	H	H
3	M	M	M	M	M	M	M
4	L	L	L	L	L	L	L
5	VL	VL	VL	VL	VL	VL	VL

4.2 Comparison of Fuzzy Result with C4.5 Algorithm

To verify the results of fuzzy logic we have used another algorithm that is used to generate decision tree. C4.5 is often termed as statical classifier.

Pseudocode and Algorithm:

C4.5 gerenrate the decision tree on the basis of set of training data using the concept of information enytropy.

The basic pseudocode is as follow:

4.2.1 Pseudocode

In pseudocode, the general algorithm for building decision trees is:

1. Check for base cases
2. For each attribute a
3. Find the normalized information gain from splitting on a
4. Let a_best be the attribute with the highest normalized information gain
5. Create a decision node that splits on a_best
6. Recurse on the sublists obtained by splitting on a_best, and add those nodes as children of node [28]

4.2.2 Implementation

J4.8 is a practical implementation of C4.5. We have used and open source utility Weka to generate node tree for our results. We have provided 25 percent of data to Weka for training data and then we used remaining data for the results node.

The results from the Weka have ignored attributes on the basis of survey results.

We have provided our modified numeric results of the survey to the Weka.

The node tree produced by the givcen data is as follow

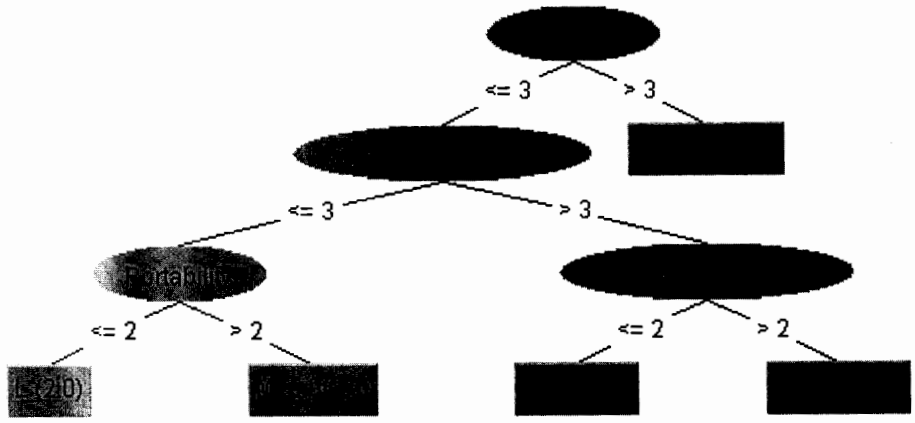


Fig. 20

4.2.3 Classifier Output from Weka

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     Modified Numeric Results
Instances:    100
Attributes:   7
              Variability
              Portability
              Maintainability
              Understandability
              Flexibility
              Scope Coverage
```

```
                Reusability

Test mode:10-fold cross-validation


=== Classifier model (full training set) ===


J48 pruned tree

------------------

Portability <= 3

|   Understandibility <= 3

|   |   Portability <= 2: L (2.0)

|   |   Portability > 2: M (5.0/1.0)

|   Understandibility > 3

|   |   Scope Coverage <= 2: L (4.0/1.0)

|   |   Scope Coverage > 2: H (16.0/6.0)


Portability > 3: H (73.0/23.0)

Number of Leaves  :   5

Size of the tree :    9


Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances          53               53      %

Incorrectly Classified Instances        47               47      %

Kappa statistic                         -0.0101

Mean absolute error                      0.2831

Root mean squared error                  0.431

Relative absolute error                 99.4127 %
```

Root relative squared error 114.8946 %

Total Number of Instances 100

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.176	0.084	0.3	0.176	0.222	0.49	M
	0.806	0.842	0.61	0.806	0.694	0.467	H
	0	0.059	0	0	0	0.411	VH
	0	0.032	0	0	0	0.673	L
Weighted Avg.	0.53	0.547	0.429	0.53	0.468	0.475	

=== Confusion Matrix ===

a	b	c	d	<-- classified as
3	13	1	0	a = M
5	50	4	3	b = H
1	14	0	0	c = VH
1	5	0	0	d = L

Chapter 5: Final Results

5- Final Results

In this section we will present the results of our study. We will present that how reusability is dependent on different attributes as we have discussed in our selected model and finally what are the results from the Fuzzification of the survey data.

We have given modified our survey results and then applied Fuzzy Logic on them and defuzzification results in the final results of our study. Applying Fuzzy Logic we have consolidated results of reusability depending on all attributes but below we have discussed dependency on these attributes separately and finally a consolidated result.The results and conclusion based on our study are as follow:

5.1 Variability and Reusability:

Variability is capability of a component to have different variants with unique capabilities. High variability increases the reusability.

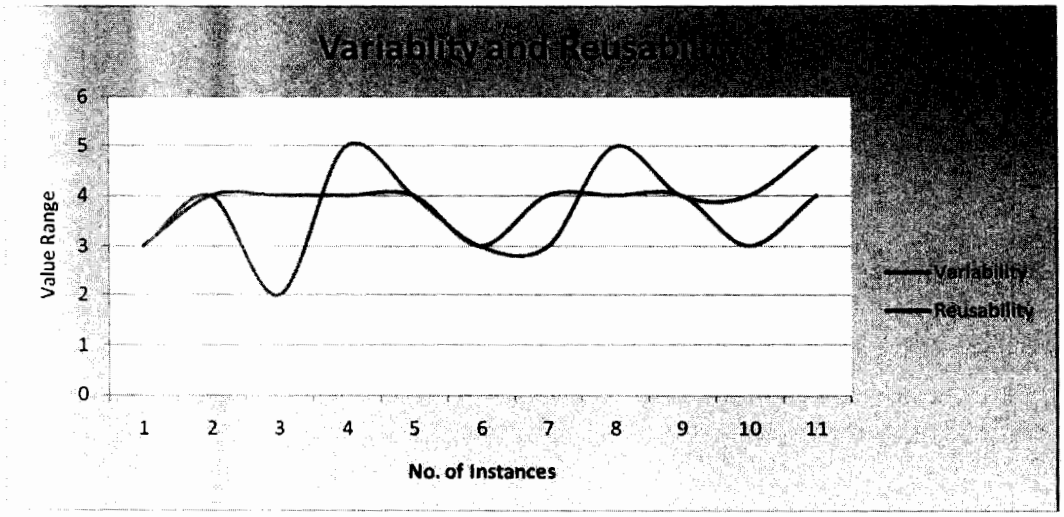


Fig. 21

Attri		
High Variability	3.9 – 5.9	If Variability is high, then reusability is high
Medium Variability	3.0 - 3.9	If Variability is medium, then reusability is medium
Low Variability	1.0 - 2.9	If Variability is low, then reusability is low

Table: Variability and Reusability

5.2 Flexibility and Reusability

Flexibility is an ability to modify and operation component. So flexibility is an important factor to measure the reusability. Increased flexibility, increases reusability.

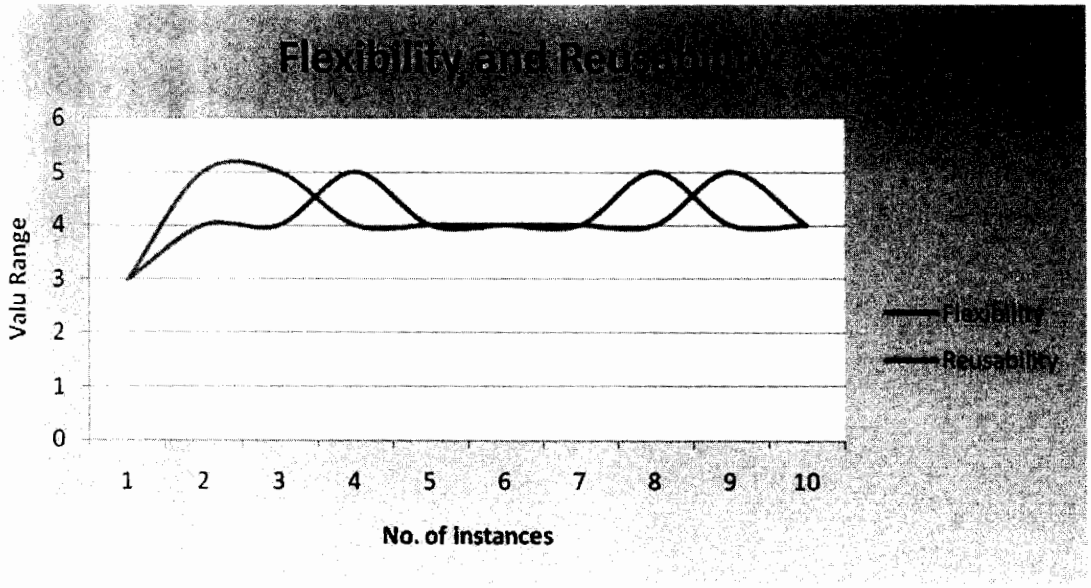


Fig. 22

Attributes		
High Flexibility	3.9 – 5.9	If Flexibility is high, Reusability is high
Medium Flexibility	3.0 - 3.9	If Flexibility is medium, Reusability is medium
Low Flexibility	1.0 - 2.9	If Flexibility is low, Reusability is low

Table: Flexibility and Reusability

5.3 Maintainability and Reusability

Maintainability is referred as effort required to locate an error and to fix it. But this modification may also be referred as addition of some features or modification of the existing or error fixing.

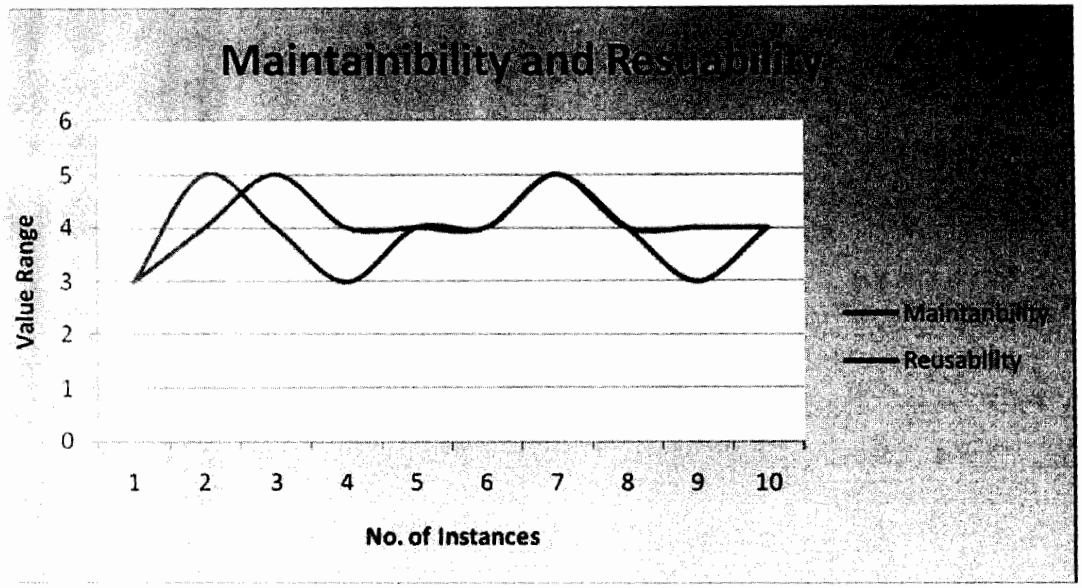


Fig. 23

Attributes		
High Maintainability	3.9 – 5.9	If Maintainability is high, reusability is also high
Medium Maintainability	3.0 - 3.9	If Maintainability is medium, reusability is also medium
Low Maintainability	1.0 - 2.9	If Maintainability is low, reusability is low

Table: Maintainability and Reusability

5.4 Understandability and Reusability

Understandability is the capability of the a software component to which a user can understand that what it is made for or what is its use and under hat condition it can be used for certain tasks.

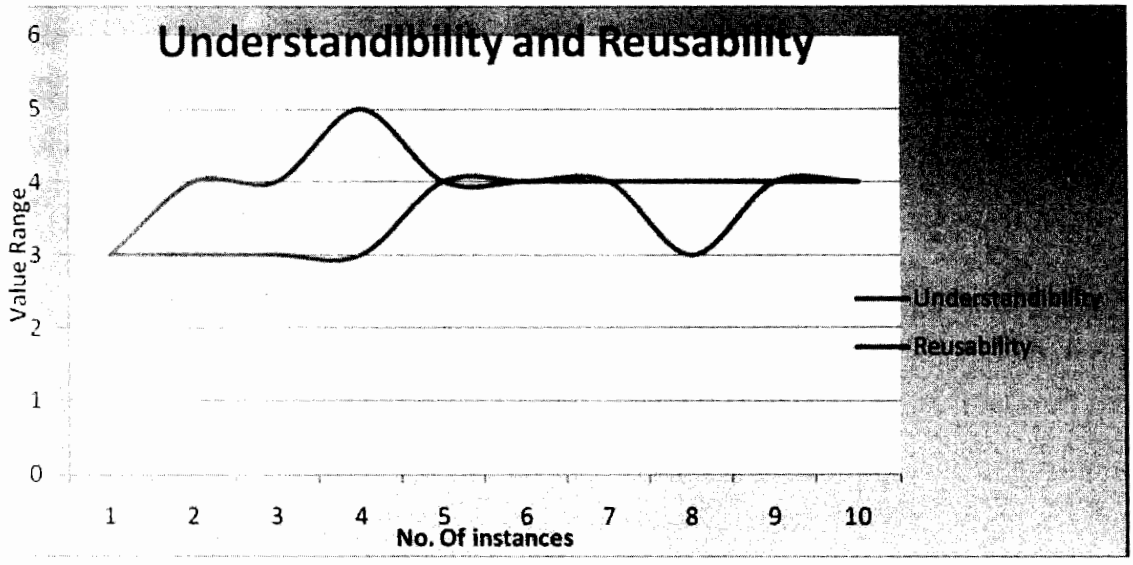


Fig. 24

Attributes		
High Understandability	3.9 – 5.9	If Understandability is high then Reusability is also high
Medium Understandability	3.0 - 3.9	If Understandability is medium then Reusability is also medium
Low Understandability	1.0 - 2.9	If Understandability is low then Reusability is also low

Table: Understandability and Reusability

5.5 Portability and Reusability

Portability is capability of a software product to be moved from one environment to another

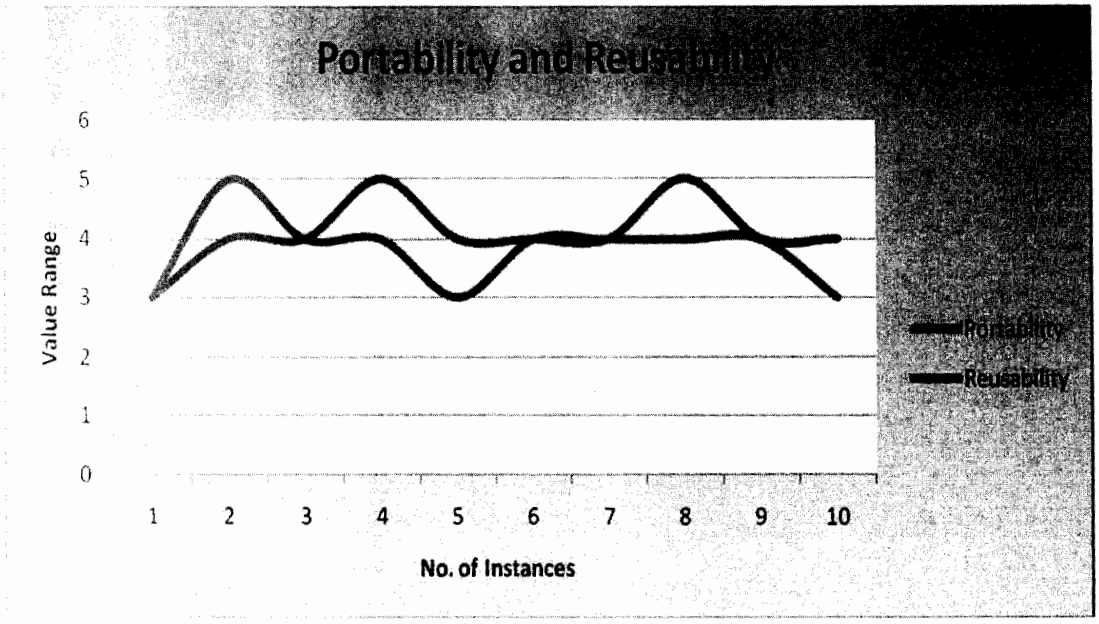


Fig. 25

Attributes		
High Portability	3.9 - 5.9	If Portability is high
Medium Portability	3.0 - 3.9	If Portability is medium
Low Portability	1.0 - 2.9	If Portability is low

Table: Portability and Reusability

5.6 Scope Coverage and Reusability

Scope Coverage is the attribute that measures the number of features provided by the component against the total number of features in product line scope. A component providing more common features will have increased reuse potential than the component supporting fewer features.

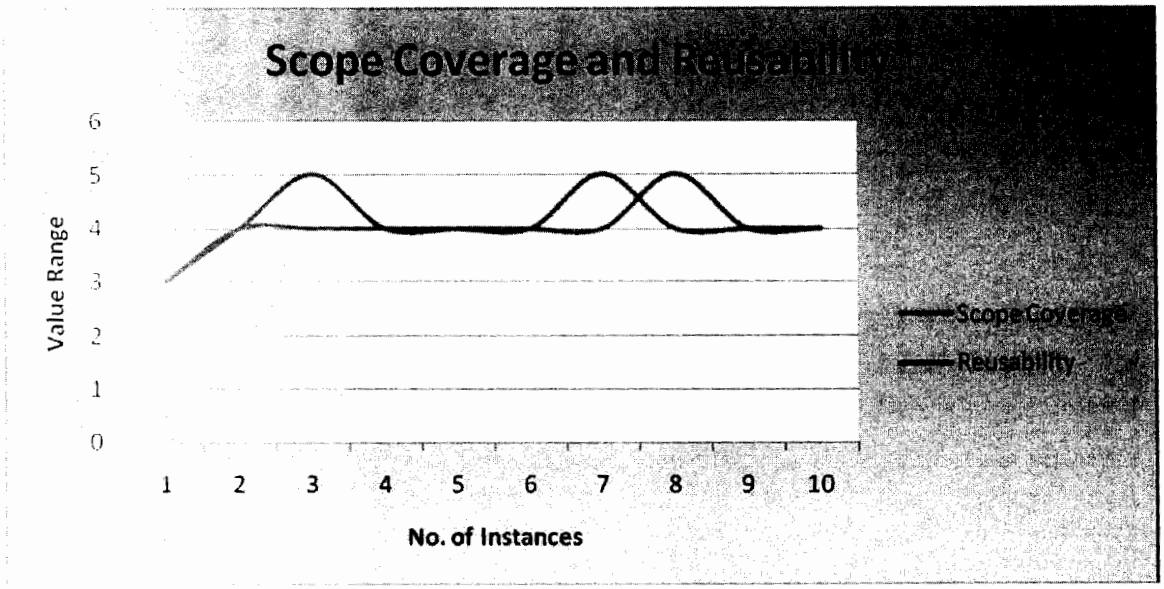


Fig. 26

Attributes		
High Scope Coverage	3.9 – 5.9	If Scope Coverage is high then Reusability is also high
Medium Scope Coverage	3.0 - 3.9	If Scope Coverage is medium then Reusability is also medium
Low Scope Coverage	1.0 - 2.9	If Scope Coverage is low then Reusability is also low

Table: Scope Coverage and Reusability

Chapter 6: Conclusion and Future Work

6- Conclusion and Future Work

In this section we will discuss our conclusion and space for future work.

6.1 Conclusion:

Software reusability is a major cost cutting technique that cab save time and cost of the software development. Re-Inventing the wheel again and again is the wastage of resources and may results in the failure of the project.

From the results of our survey, there is a need of software reusability model and Aspect Oriented Software Development focusing on reusability as cross cutting concerns depending on different non functional attributes can help for such a model. We have visualized the results of survey then their analysis using Fuzzy Logic that depending on non functional requirements enhancement, reusability can also be enhanced. In the whole study we have focused on non functional attributes mostly cross cutting concerns that can help to improve the reusability to much more extent. There are different models already proposed and we have verified one of those models applying Fuzzy Logic. It means that the market needs of reusability can be fulfilled using these techniques.

6.2 Recommendations for Future Work

Recommendation for the future work are validation of our results using these proposed and verified model performing different experiment and formulating the result of enhanced Software Reusability with respect to Aspect Oriented Software Development. Flexibility, Scope coverage, Portability, Understandability, Variability and flexibility are identified as the reusability attributes for core assets.

This will help the software industry, specially working on software product line.

7- *References*

1- Using Aspect Oriented Software Architecture for Enterprise Systems Development

Pawan Kumar Verma ,Deepak Dahiya,Pooja Jain IEEE 2010

2- Aspect-oriented Programming for MVC Framework Hui Li, Mingji Zhou, GuiJun Xu.

Lingling Si IEEE 2008

3- Briefly Introduced and Comparatively Analysed Ali Seyfi and Ahmed Patel, IEEE 2010

4- Web Software Evolution by Aspect-oriented Adaptation Engineering Matthias

Niederhausen, Zoltain Fiala, Norbert Kopcsek, Klaus Meissner IEEE 2007

5- Characterising Faults in Aspect-Oriented Programs: Towards Filling the Gap between

Theory and Practice Fabiano C. Ferrari, Rachel Burrows, Ot´avio A. L. Lemos, Alessandro

Garcia and Jos´e C. Maldonado IEEE 2010

6- On the Need of Architectural Patterns in AOSD for Software Evolution M. Pinto, L. Fuentes, J. A. Valenzuela, P. F. Pires, F. C. Delicato, E. Marinho IEEE 2009

7- Process Patterns for Aspect-Oriented Software Development Massood Khaari, Raman Ramsin IEEE 2010

8- Using AOP to Ensure Component Interactions in Component-Based Software Jingang Zhou , Yong Ji , Dazhe Zhao 1, 3 , Jiren Liu IEEE 2010

9- Using aspects for testing nonfunctional requirements in object-oriented systems Salam Farhat, Greg Simco, Frank J. Mitropoulos IEEE 2010

10- A Method for Modeling Aspect-Oriented Dynamic Software Architecture Ling Wang , Guangquan Zhang, Jihan Zhu , Jianfeng Wu IEEE 2010

11- Aspect-Oriented Software Development in Practice: Tales from AOSD-Europe Awais Rashid, Thomas Cottenier, Phil Greenwood, and Ruzanna Chitchyan, Regine Meunier, Roberta Coelho, Mario Südholt, Wouter Joosen, IEEE 2010

12- Aspect Weaving in MDA for Ontology Modelling Farheen Siddiqui ,M Afshar Alam ICEEA 2010 Thesis Proposal: Optimization of Reusability model for Aspect Oriented Software Development (AOSD)

13- Aspect-Oriented Modeling and Mapping Driven by Model Driven Architecture Jingjun Zhang,Yuejuan Chen, Yang Zhang,Hui Li IEEE 2009

14- Understanding The Aspects From Various Perspectives in Aspects-Oriented.Software Reverse Engineering ZHANG Ping, SU Yang IEEE 2010

15- A Proposed Reusability Attribute Model for Aspect Oriented Software Product Line Components Fazal-e-Amin, Ahmad Kamil Mahmood, Alan Oxley IEEE 2010

16- On the Reuse and Maintenance of Aspect-Oriented Software: An Assessment Framework Cláudio Sant'Anna , Alessandro Garcia , Christina Chavez , Carlos Lucena , Arndt von Staa

17- Towards an Aspect-Oriented Approach to Improve the Reusability of Software Process Models Rodrigo Quites Reis, Carla Alessandra Lima Reis ,Heribert Schlebbe .Daltro José Nunes

18- Aspect-Oriented Software Development beyond Programming Awais Rashid ,Alessandro Garcia ,Ana Moreira ACM 2006

19- Assessment of Reusability in Aspect-Oriented Systems using Fuzzy Logic N.W. Nerurkar, Avadhesh Kumar, Pallavi Shrivastava ACM 2010

20-ISO/IEC, "Standard 9126 Part 1,2,3," in 2001.

21- J. J. E. Gaffney, "Metrics in software quality assurance," in Proceedings of the ACM '81 conference ACM, March 1981, pp. 126-130.

22- R. S. Pressman, Software Engineering: A Practitioner's Approach, 5th ed.: McGraw-Hill, 2001.

23- J. Sanders and E. Curran, Software Quality: A Framework for Success in Software Development and Support 1st ed., 1995.

25-http://www.mathworks.com/help/toolbox/fuzzy/fp72.html

26-http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/sbaa/article1.html

27-Classification of Data Based on a Fuzzy Logic System, Mohammadian, M. , Computational Intelligence for Modelling Control & Automation, 2008 International Conference on IEEE 2008

28-S.B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, Informatica 31(2007) 249-268, 200