

**DEVELOPMENT OF DESIGN OF EXPERIMENT (DOE) SOFTWARE, FOR  
2<sup>K</sup> FACTORIAL DESIGN (WHERE K = 2 and 3).**

**Oleh:**

**Mohd Salzahrin Bin Mohd Hamzah**

**Disertasi ini dikemukakan kepada  
UNIVERSITI SAINS MALAYSIA**

**Sebagai memenuhi sebahagian daripada syarat keperluan untuk ijazah dengan  
kepujian**

**SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN ELEKTRONIK)**

**Pusat Pengajian Kejuruteraan  
Elektrik dan Elektronik  
Universiti Sains Malaysia**

**MAY 2006**

## **ABSTRACT**

Design of Experiment is a technique for understanding a process, studying the impact of potential variable of factors affecting the process and thereby providing the idea and insight for continuous quality improvement possibilities. Factorial design is one of the Design of Experiment methods to design an experiment where we have more than two factors. This thesis will show the usage of DOE and the statistical methods to run the experiment for factorial designs. The C++ Builder software program is also one of the tools in order to build this software. Research on DOE is very helpful in order to understand more on the topic so that this software will meet the objective sooner after the software is ready to use. With this software, it will help student in engineering courses to understand more about DOE especially on factorial design. Moreover, this software will be the basic tool for them to start with.

## **ABSTRAK**

Rekabentuk eksperimen adalah suatu teknik yg digunakan untuk memahami sesuatu proses, mengenalpasti kesan-kesan pembolehubah bagi sesuatu factor yang berpotensi untuk memberi kesan pada sesuatu proses maka dengan ini akan dapat memberikan sebuah idea dan gambaran awal tentang kemungkinan-kemungkinan peningkatan kualiti secara berterusan. Rekabentuk faktor adalah salah satu kaedah rekabentuk eksperimen untuk merekabentuk sebuah eksperimen yang memerlukan lebih dari dua factor. Tesis ini akan menunjukkan penggunaan DOE dan kaedah statistic untuk menjanakan eksperimen untuk rekabentuk faktor. Program Perisian C++ Builder adalah merupakan salah satu alat untuk membina perisian yang akan dibina ini. Kajian terhadap rekabentuk eksperimen memberikan banyak pertolongan dalam erti kata untuk memahami dengan lebih terhadap topik tersebut supaya perisian yang akan dibina akan dapat memenuhi objektif sebaik sahaja perisian ini dapat digunakan. Menggunakan perisian ini dapat membantu pelajar kejuruteraan untuk lebih memahami tentang rekabentuk eksperimen terutamanya tentang rekabentuk faktor. Lebih-lebih lagi, perisian ini akan menjadi alat asas untuk permulaan bagi mereka.

## **ACKNOWLEDGMENT**

A special and warm thanks to Dr. Zalina Abd Aziz who is my supervisor for my Final Year Project. She is the one who assist me towards the days so that I can finally finish my thesis and completion of my software. Moreover, I don't have any information about Design of Experiment in the first place, but with her advices, she told me to read so many books which it really helps me in order to understand the DOE and Factorial Design.

I would like to say thanks to Dr. Kamal Zuhairi for his willingness to help me in order to understand the C++ programming languages. He is also the one who taught me how to use and implement the UML to build this software. More, he gave me the C++ Builder manual to study about C++ programming. With the help of the book, I can now build my own software and for my FYP, I have to develop software about Design of Experiment for  $2^k$  Factorial Design.

Last but not least, to all of my friends who are really support me until the completion of this project and especially to my team under the same supervisor for helping me to understand more about DOE.

Mohd Salzahrin bin Mohd Hamzah

8 MAY 2006

## TABLE OF CONTENT

<b>ABSTRACT.....</b>	<b>ii</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>iv</b>
<b>TABLE OF CONTENT.....</b>	<b>v</b>
<b>LIST OF FIGURES.....</b>	<b>vii</b>
<b>LIST OF TABLES.....</b>	<b>viii</b>

### CHAPTER 1 INTRODUCTION

1.1 Introduction of the project.....	1
1.2 Objective.....	2
1.3 Scope of Research.....	3
1.4 Methodology	
1.4.1 Survey on software.....	4
1.4.2 Research on Design of Experiment (Case Study).....	4
1.4.3 Learning the C++ Builder software program.....	4
1.4.4 Case Study on UML (Unified Modeling Language).....	4
1.5 Flow Chart of the project.....	5
1.6 Structure of Thesis.....	8

### CHAPTER 2 LITERATURE REVIEW

2.1 Introduction.....	9
2.2 Introduction of Design of Experiment.....	9
2.2.1 Basic Principle.....	10
2.2.2 How to use statistical technique in experiment.....	10
2.2.3 Advantages of Design of Experiment.....	11
2.3 Introduction to Factorial Design.....	11
2.3.1 The Concept of Interaction.....	14
2.4 Introduction to $2^k$ Factorial Design.....	15
2.4.1 The $2^2$ Design.....	15
2.4.2 The $2^3$ Design.....	16
2.4.3 The ANOVA table.....	17
2.5 UML (Unified Modeling Language)	
2.5.1 Introduction.....	19

2.5.2	Use Case Diagram.....	19
2.5.3	Sequence Diagram and Collaboration Diagram.....	20
2.6	Introduction to C++ Builder Program.....	21
2.6.1	A Quick look at the C++ Builder.....	21
2.6.2	C++ Language Overview.....	23
2.7	Software Usability.....	24
2.8	Conclusion.....	25

### **CHAPTER 3 SOFTWARE DESIGN**

3.1	Introduction.....	26
3.2	Design using UML (Unified Modeling Language).....	26
3.3	The software User's Interface.....	29
3.4	Examples on the Software Design.....	33
3.5	Instruction / Manual.....	38

### **CHAPTER 4 SOFTWARE TESTING**

4.1	Introduction.....	39
4.1.1	Introduction of Unit Testing.....	39
4.1.2	Introduction of Usability Testing.....	39
4.2	Questionnaire (modified from SUMI).....	40

### **CHAPTER 5 RESULT AND DISCUSSION**

5.1	Introduction.....	43
5.2	Problems And Solutions.....	43
5.3	Result From The Feedback.....	45
5.3.1	Analysis of the Feedback Form.....	45
5.3.2	Result from the Feedback Form: The Analysis.....	46
5.3.3	Building the Histogram for Measuring the Performance from the Analysis.....	48
5.3.4	Statistical Analysis of the Feedback Form.....	49
5.3.4.1	Cause and Effect Diagram.....	50
5.4	Conclusion.....	51

## CHAPTER 6 CONCLUSION

6.1 Introduction.....	52
6.2 Recommendation.....	52
6.3 Conclusion.....	53

## REFERENCE

APPENDIX A:	PROGRAMMING CODE
APPENDIX B:	SUMI QUESTIONNAIRE
APPENDIX C:	COMPLETED QUESTIONNAIRE

## LIST OF FIGURES

Figure 2.1	A factorial experiment without interaction	14
Figure 2.2	A factorial experiment with interaction	14
Figure 2.3	The treatment combination for $2^2$ design	16
Figure 2.4	Use Case Diagram	20
Figure 2.5	The initial blank form	22
Figure 2.6	The Component Palette	22
Figure 2.7	The Object Inspector	23
Figure 3.1	Use Case Diagram for the software design	26
Figure 3.2	Sequence Diagram for the software design	27
Figure 3.3	Collaboration Diagram for the software design	28
Figure 3.4	The Design for User's Interface	30
Figure 3.5	The MenuBar of the software	30
Figure 3.6	The StringGrid (known as table for the design)	31
Figure 3.7	Panels for number of factors and the number of replicate	32
Figure 3.8	Designed Table for Two Factors with Two Replicates	33
Figure 3.9	Designed Table for Three Factors with Two Replicates	33
Figure 3.10	ANOVA table for Two Factors Factorial Design	34
Figure 3.11	ANOVA table for Three Factors Factorial Design	34
Figure 3.12	Example Design for Two Factors with Three Replicates	35

<b>Figure 3.13</b>	<b>The ANOVA result</b>	<b>35</b>
<b>Figure 3.14</b>	<b>The Main Effect Table Design</b>	<b>36</b>
<b>Figure 3.15</b>	<b>The Main Effect Plot for Factor A</b>	<b>36</b>
<b>Figure 3.16</b>	<b>The Main Effect Plot for Factor B</b>	<b>37</b>
<b>Figure 3.17</b>	<b>Interaction Plot between Factor A and Factor B</b>	<b>37</b>
<b>Figure 5.1</b>	<b>Histogram of Analysis Result from Table 5.2</b>	<b>48</b>
<b>Figure 5.2</b>	<b>The Cause-And-Effect Diagram for Question 6</b>	<b>50</b>
<b>Figure 5.3</b>	<b>The Cause-And-Effect Diagram for Question 9</b>	<b>50</b>



## LIST OF TABLES

<b>Table 2.1</b>	<b>A factorial experiment</b>	<b>11</b>
<b>Table 2.2</b>	<b>A factorial experiment</b>	<b>12</b>
<b>Table 2.3</b>	<b>Examples for <math>2^2</math> factorial design</b>	<b>14</b>
<b>Table 2.4</b>	<b>Difference notations for <math>2^3</math> design</b>	<b>15</b>
<b>Table 2.5</b>	<b>The ANOVA table for two factors factorial; fixed effect model</b>	<b>16</b>
<b>Table 2.6</b>	<b>The ANOVA table for three factors factorial; fixed effect model</b>	<b>17</b>
<b>Table 5.1</b>	<b>Hidden Objective of Questions from the Feedback Form</b>	<b>46</b>
<b>Table 5.2</b>	<b>CheckSheet for Feedback Form of the Software</b>	<b>47</b>

## CHAPTER 1

### INTRODUCTION

#### 1.1 INTRODUCTION OF THE PROJECT

Design of Experiments (DOE) is a method or a tool and is widely used in the engineering field to improve the performance of a process or in the development of a new product. The data obtained from the experiment is then been analyzed using statistical methods in order to obtain objective conclusions. Experimental Designs is not new for the industrial throughout the world. DOE method has been widely used for over 15 years for manufacturing and Services Company to upgrade their improvement in their products and services.

Many experiment in the daily life today involving two or more factors. We use factorial design for this type of experiments. Main effect is the change in response due to the change of the factor. For this software, it will be more specifically on two factors and three factors, each at two levels. This is just the basic idea towards the introduction of the software.

Development of Design of Experiment software will surely improve the usage of DOE in all sectors especially in engineering and scientific research environment. With this software, engineers will get the ability to understand how to apply DOE technique to improve the manufacturing level.

## 1.2 OBJECTIVE

The objective of this software is to give the basic idea for the users about Experimental Design technique especially on Factorial Design. More specifically is on General Factorial Design for  $2^k$  factors. The software will have the following capabilities:

- The software that is user-friendly. Users will not having any difficulty on how to use this software.
- Software that can create the ANOVA table for the users to see the result of their experiment.
- The ANOVA table will display the Sum of Square for each of the factors, Degrees of Freedom, Mean Square etc.
- The software that can calculate the factorial design for two factors and three factors.
- It can display the Main Effect Plot and the Interaction Plot. In addition, it also displays the value for each of the points in the plots.

### 1.3 SCOPE OF RESEARCH

The scope of research section will be discussing about mainly on the introduction of the Design of Experiments and the Factorial Design where the method is the  $2^k$  Factorial Design method. Design of Experiment is a tool that mainly used in huge industrial factories where the production and the process need to be inspected so that the productions will be in the highest improvements.

The Factorial Design is one of the DOE methods when there are more factors to be study of the effects. The effect of a factor is defined to be the change in response produced by a change in the level of the factor. This is called as a Main Effect; it refers to the primary factors of interest in the experiment.

The software development is mainly on the  $2^k$  Factorial Design where the software is fixed to two and three factors only, each of the factors are at two levels only. The scope is only until three factors because when we wanted to do an experiment with more than four factors, the Fractional Factorial Design will be used as the other method. This is to minimize the number of runs. The software will display the Analysis of Variance of the experiment where for two factors and the three factors ANOVA tables are Fixed Effects Model.

## **1.4 METHODOLOGY**

### **1.4.1 Survey on Software.**

On the previous semester, I did a survey on the software programs that can create software. To begin with, I choose between Visual Basic and C++ Builder. For more convenient, I use C++ Builder because I am familiar with the C++ coding.

In addition, this software is better than normal C++ software program because we can create an interface and window with C++ Builder, with the menu bar and tool bar. After we compile the project, it can be run in Windows especially Windows 2000 and Windows XP. Later on, we will discuss more about C++ Builder on the next chapter.

### **1.4.2 Research on Design of Experiment (Case Study)**

This research is on finding the definition of DOE and to study about Design of Experiment on general factorial design. Books can be found in the library and some of them have been introduced by Dr Zalina bt Abd Aziz like the “Design and Analysis of Experiment” by Douglas C. Montgomery.

### **1.4.3 Learning the C++ Builder Software Program**

In order to learn the basic concept of the C++ Builder program, I borrowed the manual guidebook from Dr. Kamal Zuhairi with the title “C++ Builder in 21 Days” by Kent Reisdorph. Inside the book, it gives an idea on how to create our own software and how to do it, step by step.

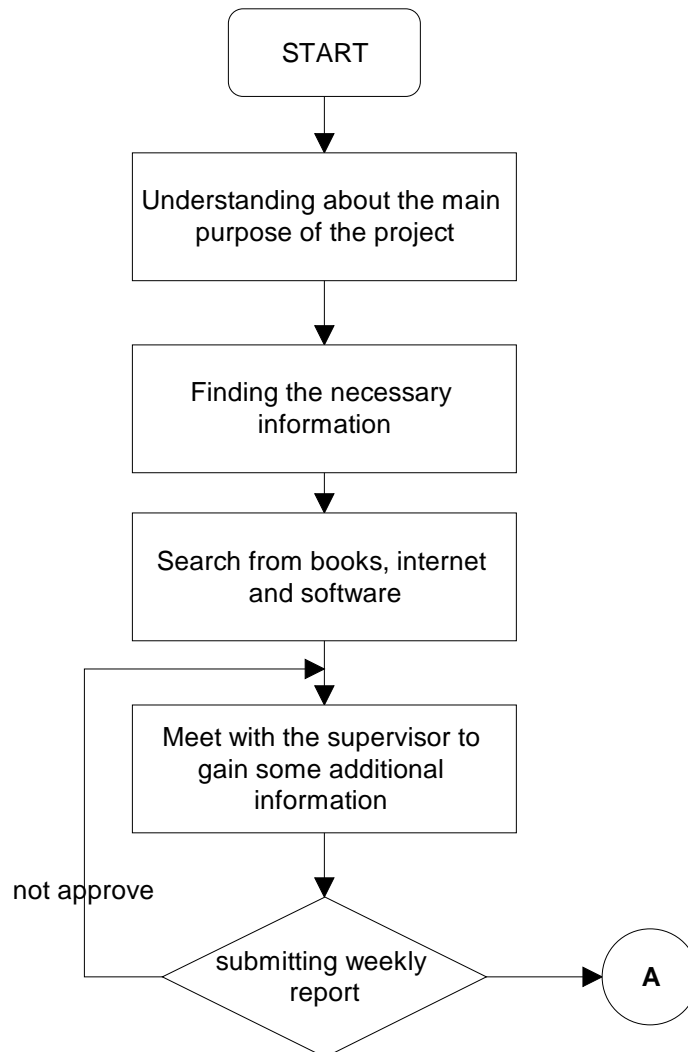
### **1.4.4 Case Study on UML (Unified Modeling Language)**

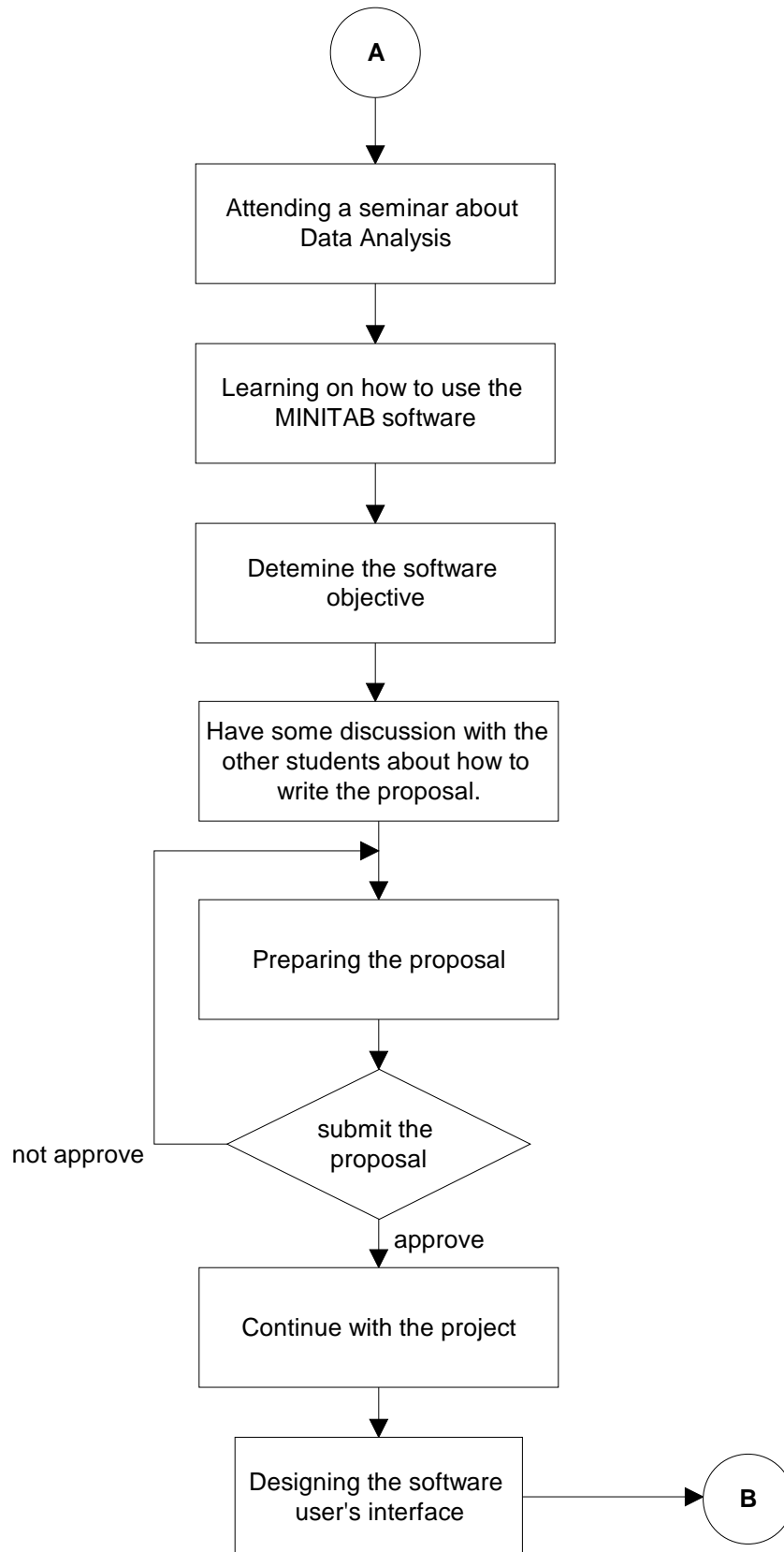
The first step in the designing software is to make an idea of the design in a graphical design. I used the UML program to illustrate the idea and the flow of my software where the UML program uses the Object-Oriented methods to visualize, specify and to document the whole of the software system.

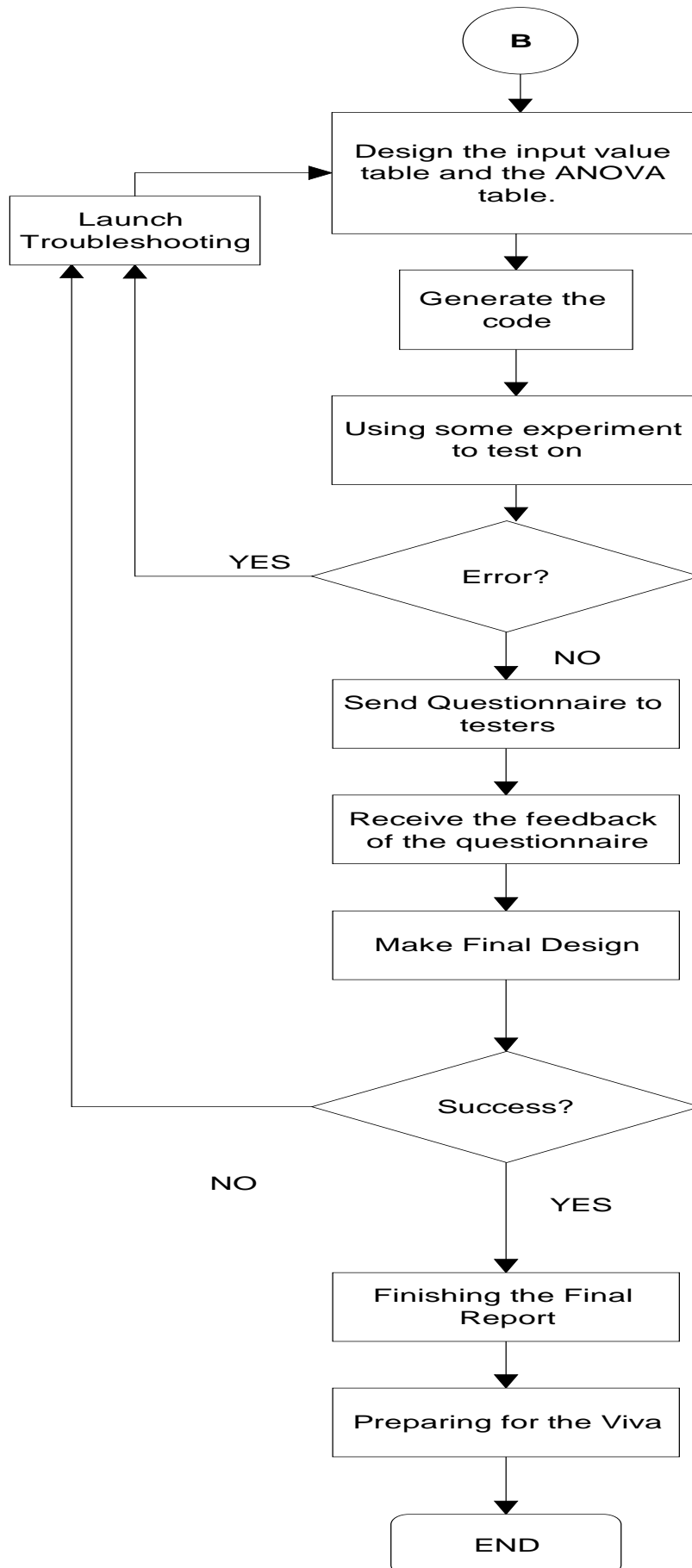
Moreover, I had learned the subject when I was in the first semester on my 4<sup>th</sup> year. Dr. Kamal Zuhairi is the one who taught me on how to use the UML to build our own software system. The program is really helping me in visualizing my idea and it become easier to implement and to construct the software program.

### 1.5 FLOW CHART OF THE PROJECT

The flow chart will describe the overall process towards the completion of the software. I begin with the process in the last semester when I first got the title of the project. Form there, I did some research about Design of Experiment and about 2<sup>k</sup> Factorial Design.









## 1.6 STRUCTURES OF THESIS

**CHAPTER 2.** In this chapter we will discuss a little bit more on the Literature Review. This will cover the research I done towards the completion of the software including designing, creating the software using C++ Builder and then programming.

**CHAPTER 3.** This chapter we will see on how the software is build where this chapter is the software design. Here I will explain in detail on the idea and the steps I take to build the software and this include the user's interface.

**CHAPTER 4.** In chapter 4, we will discuss about the testing methods that include the Unit Testing and the Usability Testing. Then I will include the questionnaire that I modified it from the SUMI questionnaires.

**CHAPTER 5.** In this chapter, I will discuss about the result of the software. Form the feedback form that I gave to some of the software tester, then from the feedback, I can determined whether my software meets the objective or not.

**CHAPTER 6.** This chapter is the conclusion chapter. In this chapter I will determine whether my software met the early objective or not. The analysis can be made by using some testing to the software and by the feedback of the questionnaires.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 INTRODUCTION

This chapter will discuss about the literature review that has been carrying out in order to develop the software. It can be list down into several topics such as about Design of Experiment, C++ Builder, Unified Modeling Language (UML) etc. We will discuss all of these in details, here in chapter 2.

#### 2.2 INTRODUCTION OF DESIGN OF EXPERIMENT

A Design of Experiment, known as DOE is a structured, organized method for determining the relationship between factors that affecting a process and the output of that process. Other definition of DOE is to conduct and analyst controlled tests to evaluate the factors that control the value of a parameter.

DOE refers to experimental methods used to quantify indeterminate measurement of factors and interaction between factors statistically as directed by mathematically systematic tables. This source is taken from [isixsigma.com](http://isixsigma.com).

Design of Experiment will give the experimenter a greater idea and understanding on the process that had been held. For over fifteen years, DOE has been applied by the manufacturing industries and services companies to improve the process, production and maintenance quality.

DOE was developed in the early 1920s by Sir Ronald Fisher at the Rothamsted Agricultural Field Research Station in London, England. Then it was brought to the US by the English practitioners after the World War II, where the chemical industries were the first to apply it. Montgomery, D.C (2001)

### 2.2.1 Basic Principles

There are seven basic principles that had been introduced by Douglas C. Montgomery in his book. To use the statistical method in designing and analyzing an experiment, it is necessary that the experimenter must know exactly what is to be studied, how the data are to be collected and how these data is to be analyzed. The recommended procedure is as follow:

1. Recognition of and statement of the problem
2. Choice of factors and levels.
3. Selection of a response variable.
4. Choice of experimental design.
5. Performing the experiment.
6. Data analysis.
7. Conclusions and recommendations.

### 2.2.2 How to use statistical techniques in experimentation

Much of the research in engineering, science and industry is empirical and makes extensive use of experimentation. Statistical methods can greatly increase the efficiency of these experiments and often strengthens the conclusions so obtained. The intelligent use of statistical techniques in experimentation requires that the experimenter keep the following points in mind. Montgomery D.C (2001)

1. Use your non-statistical knowledge of the problem.
2. Keep the design and analysis as simple as possible.
3. Recognize the difference between practical and statistical significance.
4. Experiments are usually iterative.

### 2.2.3 Advantages of Design of Experiment

The application of design of experiment techniques early in process development can result in:

- Improved process yield
- Reduced variability and closer conformance to nominal or target requirements.
- Reduced development time.
- Reduced overall cost.

Design of experimental methods also plays a major role in engineering design activities, where new products are developed and existing ones improved. Some application of experimental design in engineering design includes

- Evaluation and comparison of basic design configurations.
- Evaluation of material alternatives.
- Selection of design parameters so that the product will work under a wide variety of field conditions, that is, so that the product is robust.
- Determination of the product design parameters that impact product performance.

## 2.3 INTRODUCTION TO FACTORIAL DESIGN

Many experiments involve a study of the effects of two or more factors. In general, *factorial* designs are most efficient for this type of experiment. In each complete trial or replication of the experiment all possible combinations of the levels of the factors are investigated.

The effect of a factor is defined to be the change in response produced by a change in the level of the factor. This is called a *main effect* because it refers to the primary factors of interest in the experiment. This article was taken from Montgomery D.C (2001).

For example, let us see the data in Table 2.1 below.

**Table 2.1 A Factorial Experiment**

		Factor B	
		B1	B2
Factor A	A1	<b>20</b>	<b>30</b>
	A2	<b>40</b>	<b>52</b>

The main effect of factor A could be thought of as the differences between the average response at the first level of A and the average response at the second level of A. Numerically, this is

$$A = \frac{40 + 52}{2} - \frac{20 + 30}{2} = 21 \quad (2.1)$$

That is, increasing factor A from level 1 to level 2 causes an average response increase of 21 units. Similarly, the main effect of B is

$$B = \frac{30 + 52}{2} - \frac{20 + 40}{2} = 11 \quad (2.2)$$

If the factors appear at more than two levels, the above procedure must be modified, since there are many ways to express the differences between the average responses. This point is discussed more completely later.

In some experiments, we may find that the difference in response between the levels of one factor is not the same at all levels of the other factors. When this occurs, there is an interaction between the factors. For example, consider the data in Table 2.2.

**Table 2.2 A Factorial Experiment**

---

		Factor B	
		B1	B2
Factor A	A1	<b>20</b>	<b>40</b>
	A2	<b>50</b>	<b>12</b>

---

At the first level of factor B the A effect is

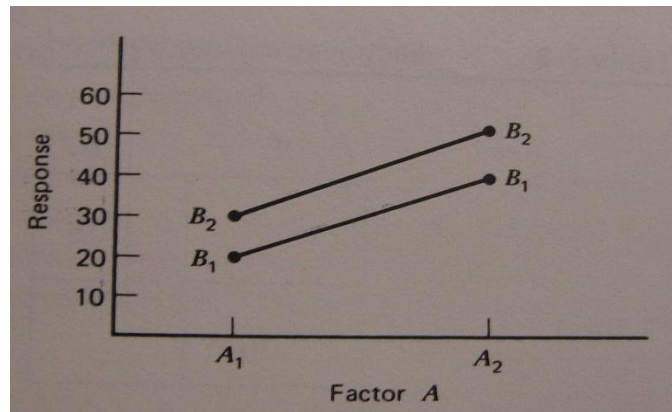
$$A = 50 - 20 = 30 \quad (2.3)$$

And the second level of factor B the A effect is

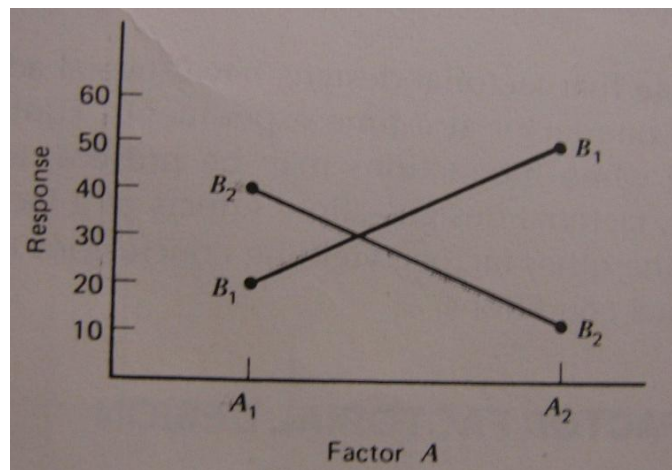
$$B = 12 - 40 = -28 \quad (2.4)$$

Since the effect of A depends on the level chosen for factor B, we see that there is interaction between A and B.

### 2.3.1 The Concept of Interaction



**Figure 2.1** A factorial experiment without interaction.



**Figure 2.2** A factorial experiment with interaction.

Figure 2.1 plots the response data in Table 2.1 against factor A for both levels of factor B. As we can see, the B<sub>1</sub> and B<sub>2</sub> lines are approximately parallel, indicating a lack of interaction between factors A and B.

Figure 2.2 plots the response data in Table 2.2. Now we can see that the lines are not parallel. This indicates an interaction between factors A and B. This type of graph is useful in interpreting significant interactions and in reporting results to non-statistically trained management. The figures were taken from the book “Design and Analysis of Experiments by Montgomery D.C (2001).

## 2.4 INTRODUCTION TO $2^k$ FACTORIAL DESIGN

The  $2^k$  is particularly useful in the early stages of experimental work, when they are likely to be many factors investigated. It provides the smallest number of treatment combinations with which  $k$  factors can be studied in a complete factorial arrangement. Because there are only two levels for each factor, we must assume that the response is approximately linear over the range of the factor levels chosen.

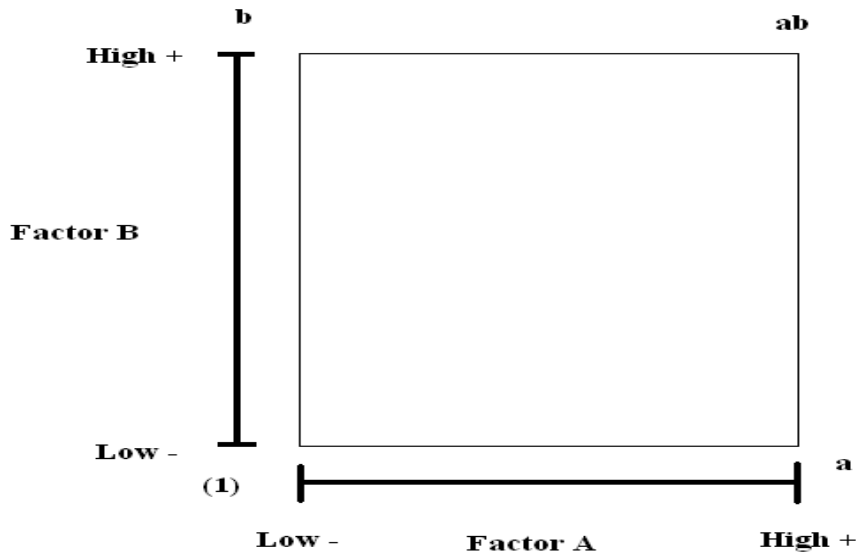
### 2.4.1 The $2^2$ Design

The first design in the  $2^k$  series is with only two factors, we might say A and B, each run at two levels. This design is called a  $2^2$  factorial design. As an example:

**Table 2.3. Example for  $2^2$  factorial designs.**

Factor		Treatment	Replicate			Total
A	B	Combination	I	II	III	
-	-	A low, B low	28	25	27	80
+	-	A high, B low	36	32	32	100
-	+	A low, B high	18	19	23	60
+	+	A high, B high	31	30	29	90





**Figure 2.3** The treatment combination for 2<sup>2</sup> design.

#### 2.4.2 The 2<sup>3</sup> Design

Suppose that three factors, A, B and C, each at two levels. The design is called 2<sup>3</sup> *factorial design* and the eight treatment combinations can be displayed as a cube geometrically. The treatment combination in standard order as (1), a, b, c, ab, ac, bc and abc.

There are actually three different notations that are widely used for the runs in the 2<sup>k</sup> design. The first is the + and – notation. The second is the use of lowercase letter labels to identify the treatment combinations. The final notation uses 1 and 0 to denote high and low factor levels.

**Table 2.4** Different notations for the 2<sup>3</sup> design.

Run	A	B	C	Labels	A	B	C
1	-	-	-	(1)	0	0	0
2	+	-	-	a	1	0	0
3	-	+	-	b	0	1	0
4	+	+	-	c	1	1	0
5	-	-	+	ab	0	0	1
6	+	-	+	ac	1	0	1
7	-	+	+	bc	0	1	1
8	+	+	+	abc	1	1	1

There are seven degrees of freedom between the eight treatment combinations in the 2<sup>3</sup> design. Three degrees of freedom are associated with the main effects of A, B and C. Four degrees of freedom are associated with interactions, one each with AB, AC, BC and one with ABC.

### 2.4.3 THE ANOVA TABLE

For **two factor** design:

$$Y_{ijk} = \mu + m_i + \beta_j + (\tau\beta)_{ij} + \epsilon_{ijk} \quad (2.5)$$

**Table 2.5 The ANOVA table for two-factor factorial; Fixed Effect Model**

Source Of Variation	Sum Of Squares (SS)	Degrees Of Freedom	Mean Square	F <sub>0</sub>
A Treatments	SS <sub>A</sub>	a - 1	MS <sub>A</sub> = $\frac{SS_A}{a - 1}$	F <sub>0</sub> = $\frac{MS_A}{MS_E}$
B Treatments	SS <sub>B</sub>	b - 1	MS <sub>B</sub> = $\frac{SS_B}{b - 1}$	F <sub>0</sub> = $\frac{MS_B}{MS_E}$
Interaction	SS <sub>AB</sub>	(a - 1)(b - 1)	MS <sub>AB</sub> = $\frac{SS_{AB}}{(a-1)(b-1)}$	F <sub>0</sub> = $\frac{MS_{AB}}{MS_E}$
Error	SS <sub>E</sub>	ab(n - 1)	MS <sub>E</sub> = $\frac{SS_E}{ab(n - 1)}$	
Total	SS <sub>T</sub>	abn - 1		

**Table 2.6 The ANOVA table for the three-factor Fixed Effects Model**

Source Of Variation	Sum Of Squares (SS)	Degrees Of Freedom	Mean Square	F <sub>0</sub>
A Treatments	SS <sub>A</sub>	a - 1	$MS_A = \frac{SS_A}{a - 1}$	$F_0 = \frac{MS_A}{MS_E}$
B Treatments	SS <sub>B</sub>	b - 1	$MS_B = \frac{SS_B}{b - 1}$	$F_0 = \frac{MS_B}{MS_E}$
C Treatments	SS <sub>C</sub>	c - 1	$MS_C = \frac{SS_C}{c - 1}$	$F_0 = \frac{MS_C}{MS_E}$
Interaction AB	SS <sub>AB</sub>	(a - 1)(b - 1)	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$F_0 = \frac{MS_{AB}}{MS_E}$
Interaction AC	SS <sub>AC</sub>	(a - 1)(c - 1)	$MS_{AC} = \frac{SS_{AC}}{(a-1)(c-1)}$	$F_0 = \frac{MS_{AC}}{MS_E}$
Interaction BC	SS <sub>BC</sub>	(b - 1)(c - 1)	$MS_{BC} = \frac{SS_{BC}}{(b-1)(c-1)}$	$F_0 = \frac{MS_{BC}}{MS_E}$
Interaction ABC	SS <sub>ABC</sub>	(a - 1)(b - 1)(c - 1)	MS <sub>ABC</sub>	$F_0 = \frac{MS_{ABC}}{MS_E}$
Error	SS <sub>E</sub>	abc(n - 1)	MS <sub>E</sub>	
Total	SS <sub>T</sub>	abcn - 1		

## 2.5 UML (UNIFIED MODELLING LANGUAGE)

### 2.5.1 Introduction

UML is a visual language for visualizing, specifying, constructing, and documenting of software system. UML is widely used in the industry .e.g. backed by leading software producers including Hewlett Packard, IBM, Microsoft, Oracle, Digital etc. Supports the entire software development lifecycle and can be used in diverse application areas. This UML is supported by many tools, and the implementation can be programming language independent. Dr.Kamal Zuhairi (2005).

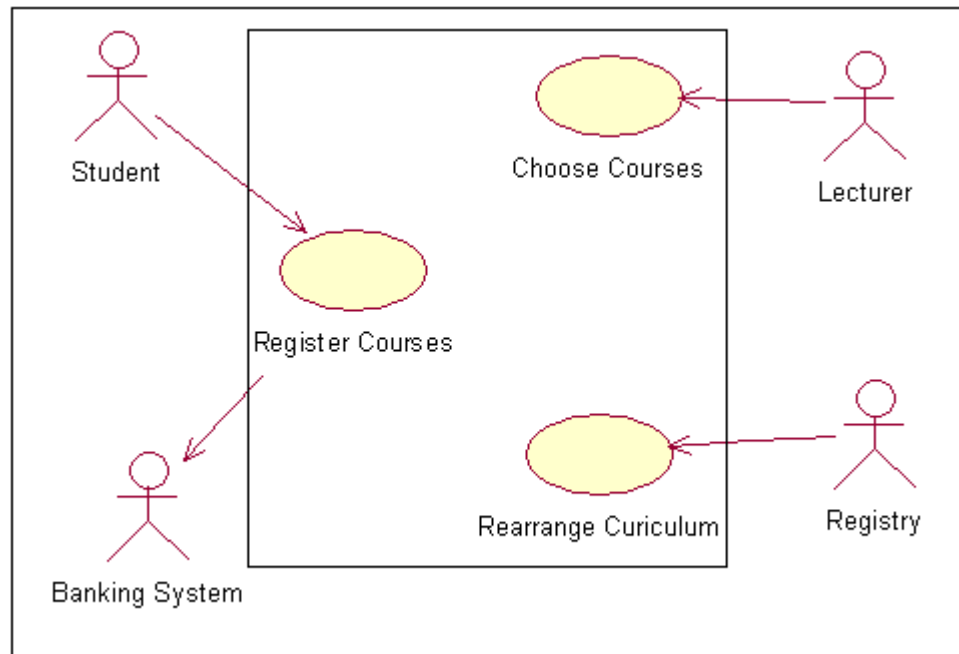
There are nine types of diagrams in UML:

- Use Case Diagram
- Sequence Diagram
- Class Diagram
- Object Diagram
- Collaboration Diagram
- State Chart Diagram
- Component Diagram
- Activity Diagram
- Deployment Diagram

The diagrams describe either the structural or the behavioral of the system.

### 2.5.2 Use Case Diagram

The Use Case Diagram shows the interaction between the actor and the system. Here, the actor can be known as someone who uses the system or something that interact directly with the system. The Diagram will represent all of the actors, Use Case and the interaction that has been classified in the system. As an example below:



**Figure 2.4 Use Case Diagrams**

### 2.5.3 Sequence Diagram and Collaboration Diagram

The Sequence Diagram shows the interaction amongst classes or the object passing the messages. The Sequence Diagram and Collaboration Diagram are also known as interaction diagrams. Each diagram provides a different view of the same interaction. The Sequence diagrams are time ordered and give focus of control, while the Collaboration diagrams may include data flow.

The Sequence diagrams illustrate how objects and actors interact with each other and sometimes used to describe scenarios. Sequence diagrams have two axes;

- Object/classes along the horizontal axis
- Lifeline (time) down the vertical axis

Objects communicate using messages. Messages are communications between objects that convey information with the expectation that something will happen. There are two passes approaches, the first pass will create messages then the second pass will converts messages to operations.

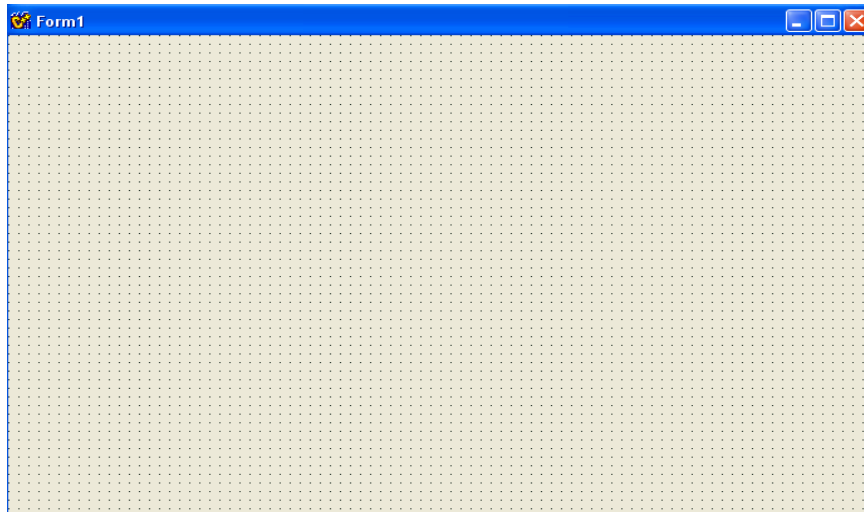
## **2.6 INTRODUCTION TO C++ BUILDER PROGRAM**

The C++ Builder is a program that gives us the ability to write C++ Windows programs more quickly and more easily than was ever possible before. We can create Win32 console applications or Win32 GUI (Graphical user interface) programs. When creating Win32 GUI applications with C++ Builder, you have all the power of C++ wrapped up in a RAD environment. It means that we can create the user interface to a program where the user interface means the menus, dialog boxes, main window, and so on using drag and drop techniques for true rapid application development. Kent Reisdorph (1998).

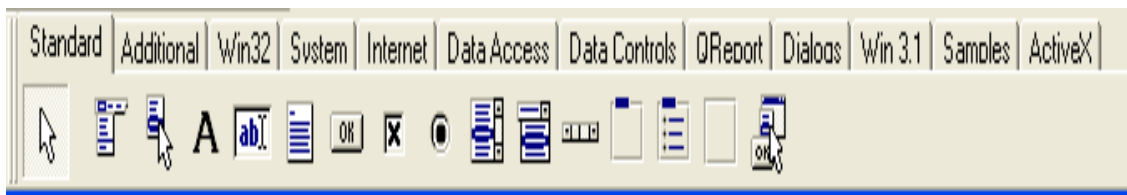
### **2.6.1 A Quick Look at the C++ Builder**

The C++ Builder IDE is divided into three parts. The top window can be considered the main window. It contains the toolbar on the left and the Component palette on the right. The toolbar gives one-click access to tasks such as opening, saving, and compiling project. The Component palette contains a wide array of components that can be drop onto our forms.

Components are text labels, edit controls, list boxes, buttons and etc. For convenience, the components are divided into groups. To place a component on the form, just simply click at the component's button in the Component palette and then click on the form where we wanted it to appear.



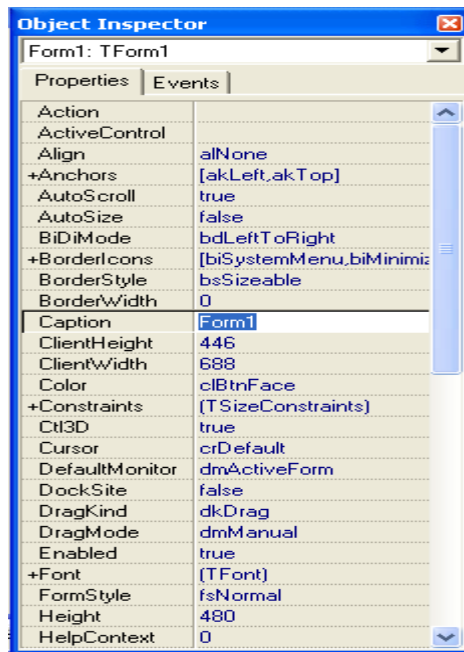
**Figure 2.5** The initial blank form.



**Figure 2.6** The Component palette

Below the toolbar and Component palette and on the left side of the screen is the Object Inspector. It is through the Object Inspector that we modify the component's properties and events. We will use the Object Inspector constantly as we work with C++ Builder. The Object Inspector has one or two tabs, depending on the component currently selected. It always has the Properties tab.

A component's properties control how the component operates. For example, changing the Color property of a component changes the background colour of that component. The list of properties available varies from component to component, although components usually have several common elements.



**Figure 2.7** The Object Inspector

## 2.6.2 C++ Language Overview

C++ is a powerful language. It enables us to do things that are not possible in other languages. As it is true in most of life, that kind of power does not come without responsibility. C++ enables us to take advantages of object-oriented programming (OOP) to its fullest. OOP is not just a buzzword. It has real benefits because it enables you to create objects that can be used in the current program and reused in future programs.

An object reveals to the user (the programmer using the object) only as much of itself as it needed to simplify its use. OOP enables us to take a modular approach to programming, thus keeping us from constantly re-inventing the wheel. C++ Builder programs are OOP oriented because of C++ Builder's heavy use of components. After a component is created, it can be reused in any C++ Builder program.



A component can also be extended by inheritance to create a new component with additional features. Components hide their internal details and let the programmer concentrate on getting the most out of the component.

## 2.7 SOFTWARE USABILITY

There are several questionnaire based approaches to assessing the usability of the software, and one such usability questionnaire (**SUMI**) developed by the Human Factors Research Group (**HFRG**) at University College Cork is discussed is here. This information was taken from O, Reagen (2002).

The SUMI methodology and the WAMMI web based tool are a questionnaire-based approach for accessing usability. Usability is a multi dimensional concept with several properties associated with each dimension of usability. The SUMI methodology list five dimensions of usability to be measured and these dimensions are related to users' expectations and attitudes to the computer system being evaluated.

- **Helpfulness.** This measures the degree to which the software is self-explanatory as well as adequacy of help facilities and documentation.
- **Control.** This measures the extent to which the user feels in control of the software as opposed to being controlled by the software.
- **Learn-ability.** This measures the speed of learning and ease of mastering the software system or new features.
- **Efficiency.** This measures the extend to which the users feel the software assists them in their work.
- **Affect.** This measures the degree to which users like the computer system .ie likeability or the general emotional reaction to the software.