VARIATION RISK MANAGEMENT

PROJECT WORK DONE AT
ASTERISKSDLINKS PVT. LTD.,
CHENNAI.

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF
M.Sc [APPLIED SCIENCE] SOFTWARE ENGINEERING
OF BHARATHIAR UNIVERSITY, COIMBATORE.

SUBMITTED BY

B.C. SANTHOSH KUMAR
REG NO. **9937S0088**

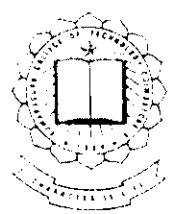
UNDER THE GUIDANCE OF

External Guide

Mr. Deepak
Asterisksdlinks
Chennai

Internal Guide

Mrs.S. Devaki
Dept. Of CSE.
Coimbatore – 6



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE – 641 006
MAY 2002 – AUG 2002

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)
COIMBATORE – 641 002
SEPTEMBER – 2002

CERTIFICATE

This is to certify that the project entitled

VARIATION RISK MANAGEMENT

DONE BY

B.C. SANTHOSH KUMAR
REG NO. **9937S0088**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
M.Sc [Applied Science] SOFTWARE ENGINEERING
OF BHARATHIYAR UNIVERSITY

Professor and HOD

26/9/2002
Internal Guide

Submitted to University Examination held on 26.9.2002

Internal Examiner

External Examiner

Asterisks* dlinks

Date: 20.09.2002

CERTIFICATE

This is to certify that **Mr. B.C. SANTHOSH KUMAR,** undertook the project **"Variation Risk Management"** in our organization from June 2002 to September 2002.

The part of this project he has completed embodies the original work done of the said module in the partial fulfillment of the requirement of the degree **M.Sc [Applied Science] Software Engineering**.

Mr. B.C. Santhosh Kumar has worked on this project with extreme sense of ownership and dedication. During the above period his conduct and character were good.

For Asterisksdlinks

Director – Executions

81, Lakshmipuram, 6th Street, B.R. Puram, Peelamedu, Coimbatore - 641 004
19, Income Tax Colony, 1st Street, Adambakkam, Chennai - 600 088. Ph: 91-44 [illegible]
3, Sundial Avenue, New Hampshire North London, UK. Ph: 0044 781 [illegible]

ACKNOWLEDGEMENT

To add meaning to the perception, it is my indebtedness to honor a few who had helped me in this endeavor, by placing them on record.

With profound gratitude, I am extremely thankful to **Dr.K.K.Padmanaban B.Sc (Eng), M.tech, Ph.D.**, Principal, Kumaraguru Collage of Technology, coimbatore for providing me an opportunity to undergo the MSc [APPLIED SCIENCE SOFTWARE ENGINEERING] course and thereby this project work also.

I extend my heartfelt thanks to my CSE department head **Prof.Dr.S.Thangasamy B.E (Hons), Ph.D.**, for his kind advice and encouragement to complete this project successfully.

It's my privilege to express my deep sense of gratitude and profound thanks to **Mr Deepak, Project Manager**, ASTRICKS Software Pvt ltd, Chennai for having allowed me to do my project work in his esteemed team and for helping me in all means in successful completion of this project work.

Gratitude will find least meaning without thanking my guide as well as course coordinator **Mrs. S. Devaki B.E, M.S.** for the valuable guidance and support throughout my project.

Words are boundless for me to express my deep sense of gratitude and profound thanks to all my associates at Astricks, for all their kind guidance and encouragement towards my project work.

My gratitude is due to all staff members of CSE department, my parents and all my friends for their moral support and encouragement for successful completion of my project.

SYNOPSIS

This project entitled "Variation Risk Management" has been developed for Asterisksdlinks. The main aim of this VRM Kit is a reduce manually made calculation based on various risk factors in a R & D Department.

On execution, provisions are made for the user to either view the exiting analysis or create new analysis. This project undergoes calculations like root sum square involving in values of lower bound, upper bound, mean and standard deviation of the selected factors. The CPK graph reports the user about the selected parameters by practical values like the cost and noise matrices and gives the user about the variations in the analysis. It also reports to the user the response graph depicting the variations in the analysis and the optimum values that may be favourable by calculating different values like SN Ratio, response time using the particular formulas.

The VRM Kit provides back tracking facility so that the user can change the values or view the values. The fixed analysis types are encrypted so that no unknown user can change the values. The existing analysis files are also secured, as they are stored using the modules in Windows files.

CONTENTS

1. INTRODUCTION

The various characteristics of the project is depicted here including the details of the organization, platform on which the project is developed along with the hardware configuration.

1.1 ORGANIZATION PROFILE

The project "Variation Risk Management" was done for Asterisksdlinks. Asterisksdlinks is a complete business solution company started during mid 2001, which deals with technical difficulties in organizations. Solutions involve the field Information Technology providing solutions. It proves its growth by establishing its office at North London and Development Centre at Chennai and Coimbatore.

The major clients of Asterisksdlinks includes

Indian Product Corporation	-	Chennai
Roots Industries	-	Coimbatore
Indian Bank	-	Coimbatore
Super Machineries	-	Coimbatore

1.2 PROJECT OVERVIEW

This project "Variation Risk Management" is developed using Visual Basic 6.0. It is a stand-alone project. The VRM Kit is expected to be flexible and allow provision for backtracking. The security of existing files is also maintained using Window based files. The fixed factors are encrypted so that unknown user cannot interfere with the values. The main objective of the proposed system is to develop the VRM Kit, a user-friendly and accuracy of calculations.

1.3 SYSTEM CONFIGURATION

1.3.1 Hardware Configuration

Processor	:	Intel 486 and Above
Printer	:	Dot Matrix or Deskjet / Inkject or Laser
RAM	:	32 MB & Above
Hard Disk	:	2 GB & Above

1.2.2 Software Specification

Operating System	:	Windows 9x Windows NT / 2000 / XP
Front End	:	Visual Basic 6.0

1.4 TECHNICAL QUICK REFERENCE

ABOUT VISUAL BASIC 6.0

The Visual Basic language is quite powerful. If we imagine a programming task, it can probably be accomplished using Visual Basic. It is a Graphical User Interface (GHUI) Application.

OLE Automation:

Using OLE Automation in Visual Basic, we can "Borrow" the functionality of other applications by controlling their objects from within our Visual Basic Application. If the object is an OLE Automation object, we can use its properties and methods in code. Each OLE automation object is a building block. We can use in code to collect and expose data and functions from other applications in a way that means sense in our application. For example, we can built an application that uses Microsoft Excel as a calculation Engine.

Enhanced Development Environment

We can sue context or popup menus for forms, controls, code, and attach add-ins for source code control and other features. With the professional edition we can create or own add-ins.

Properties Procedures

Property procedures allow to add custom properties tc form standard and class modules and to execute code when the property to a form. When inverted is set to true, the code in the associated property procedure invokes an application to invert a bitmap application on the form.

Object Browser

The object browser presents a hierarchical display of all the classes, properties and methods available to our applications from Visual Basic and other OLE Components. This allows us to easily select object and cut and paste them into our code.

Enterprise Edition

The Enterprise edition allows professionals to create robust distributed applications in a team setting. It includes back office tools such as SQL Server, Microsoft transaction server, internet information server, Visual Source safe, SNA Server and more. Printed documentation provided with the Enterprise edition includes the Visual Studio Enterprise features book plus Microsoft Developer Network CDs containing full online documentation.

Bound Data Controls

Bound data controls are designed to permit us to create data access applications that require little or no coding.

There are ten data aware controls in addition to the data control provided with Visual Basic.

Code Modules

Code in Visual Basic is stored in modules. There are kinds of modules:

Form, Standard and class modules. There are two modules used in this project namely, Forma and Standard modules.

Form Module

Form modules are the foundation of most Visual Basic applications. They can contain procedures that handle events, general procedures and form-level declarations of variables, constants, types and external procedures. The code we write in a form module is specific to the particular application to which the form belongs; it might also reference other forms or objects within that application.

Standard Module

Standard modules (.BAS file extension) are containers for procedures and declarations commonly accessed by other modules within the application. They can contain global (available to whole application) or module-level declarations of variables, constants, types, external procedures and global procedures. The code that we write in a standard module isn't necessarily tied to a particular application; if we're careful to reference forms or controls by name, a standard module can be reused in many different applications.

2. SYSTEM STUDY

The system study focuses on the stages of the system development life cycle. A complete understanding of the software requirement is essential to the success of a software development effort. The requirement area task is a process of discovery, retrievement, modeling and specification.

2.1 REQUIREMENT ANALYSIS

Requirement Analysis deals with understanding the user's requirements within the frame work of the organization's objectives and the environment in which the system is being installed.

It also deals with

- The software to be used.
- Evaluating the system concept for feasibility
- Defining the overall system objective

2.2 EXISTING SYSTEM – AN OVERVIEW

The present system involves every operation to be done manually. There are certain analysis and activation have to be carried out while developing a product, the R & D Department of a

company takes care of the analysis and implementations. There comes variations on different factors which they are considering and the variations have to be controlled. The optimum have to be calculated. Analysis are of different types. Analysis differ for each product. The factor for each product and product varies. The VRM does the job.

Some of the Horn and Analysis Types are as follows:

Horn Types:

- Vibrasonic
- Wolfhonda
- Air Nostler
- Air Blow mash
- Hakkit
- Hellasoft
- Hellahard
- Fire
- Blowjack

Analysis Types:

- Shatter
- Vibration
- Output
- Stretch
- Holdup

2.2.1 Limitations of Existing System

The existing system is bound by various difficulties such as follows

- Every operation needs manual interpretation
- It requires as much as 16 employees for its working
- Time consuming
- No history is maintained for future analysis
- The calculations are very much tedious

2.3 PROPOSED SYSTEM

The proposed system is developed to be user friendly and its maintains security of the details. This project "Variation Risk Management" is developed using Visual Basic 6.0. It is a stand-alone project. The VRM Kit is expected to be flexible and allow provision for backtracking. The security of existing files is also maintained using Window based files. The fixed factors are encrypted so that unknown user cannot interfere with the values.

2.3.1 Objectives

The main objective of the proposed system is to develop the VRM Kit, a user-friendly and accuracy of calculations. The VRM Kit should provide the user the following:

- Options of either viewing the existing analysis or to create a fresh analysis files are to be kept secured using Window Based files with the help of modules in Visual Basic.
- The root sum square (RSS) of the values of lower bound, upper bound, mean and SD of the selected parameters is to be calculated using appropriate formulae. This RSS value is to calculate the CPK (Cost Per Kelvin) value which decides the parameters to be involved in further analysis.
- The CPK graph reports the user the selected three parameters by calculation function values.
- The practical values Cost and Noise matrices of the selected factors are collected from the user.
- Using the matrices specification, VRM Kit calculates the values like SN Ratio, Response time.
- With the resultant value the response graph is reported to the user indicating the variations in the analysis.
- The maximum and minimum optimum for each factor is selected and reported.
- The best optimum for each factor is also recommended to the user.

2.3.2 Benefits

The VRM Kit can successfully yield a lot of flexibilities like the following:

- Fully automated system
- Single operator
- Rapid
- Accurate
- History is maintained which is used for future reference
- High user interface
- Adaptable to networks
- Flexible
- Can be enhanced
- Secured

3. SYSTEM ANALYSIS

System study is nothing but "Study of what the system does". It focuses on the functional and behavioral model of the problem. Implementation of the same, in the project has been dealt with. Structured analysis is a model building activity. This technique focuses in the functions and that are necessary to solve the problem. The tools used in the Structured Analysis are:

- Context Analysis Diagram
- Architectural Flow Diagram

3.1. ENVIRONMENTAL MODEL

The environmental Model defines the interface between the system and the rest of the universe, (i.e., the environment in which the system exists). The event list and Content Analysis Diagram (CAD) for the proposed system is developed. The context analysis diagram is shown in Fig.3.1.

The entities involved in the VRM are USER and MANAGEMENT. The system collects various factors, analysis details and matrices specifications from the USER. The reports generated from the collected details are given to the entity MANAGEMENT.

3.2 THE TECHNIQUE OF ARCHITECTURE FLOW DIAGRAMMING

This section describes in detail the architecture flow diagramming technique. It is intended to serve as a handbook to guide the reader in developing architecture flow diagramming skills.

Definition

Architecture flow diagramming is a means of representing a system at any level of detail with a graphic network of symbols showing architecture flows, data stores, data processes, and data sources / destinations.

Purpose / Objective

The purpose of architecture flow diagrams is to provide a semantic bridge between users and systems developers. The diagrams are:

- Graphical, eliminating thousands of words;
- Logical representations, modeling WHAT a system does, rather than physical models showing HOW it does it!
- Hierarchical, showing systems at any level of detail; and
- Jargon less, allowing user understanding and reviewing.

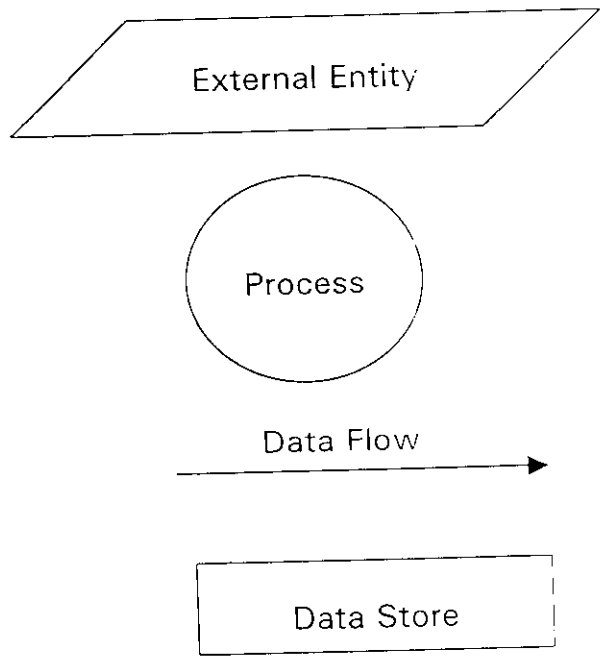
The goal of architecture flow diagramming is to have a commonly understood model of a system. The diagrams are the basis of structured systems analysis. Architecture flow diagrams are supported by other techniques of structured systems analysis such as data structure diagrams, data dictionaries, and procedure-representing techniques such as decision tables, decision trees, and structured English.

Architecture flow diagrams gave the objective of avoiding the cost of:

- User / developer misunderstanding of a system, resulting in a need to redo systems or in not using the system.
- Having to start documentation from scratch when the physical system changes since the logical system, WHAT gets done, often remains the same when technology changes.
- Systems inefficiencies because a system gets "computerized" before it gets " systematized".
- Being unable to evaluate system project boundaries or degree of automation, resulting in a project of inappropriate scope.

3.2.1 Components of Architectural Flow Diagram

Architecture flow Diagram are composed of the some basic symbols shown below



External Entities

The External Entity symbol represents sources of data to the system or destinations of data from the system. They determine the system boundary. They are external to the system being studied and are often beyond the area of influence of the developer.

Process

The Process symbol represents an activity that transforms or manipulates the data (Combines, Reorders, Converts, etc.). Datacoming into a process must be "Worked on" or transformed in some way. Thus, all processes must have inputs and outputs. In some (rare) cases, data inputs or outputs will only be shown at more detailed levels of the diagrams. Each process in always "running" and ready to accept data. They are represented by rounded corner rectangle.

Architectural Flows

The Architectural Flow symbol represents movement of data. It should represent only data, not control. They are represented with a line with an arrowhead on one end.

Data Stores

The Data Store symbol represents data that is not moving (delayed data at rest). They are generic for physical files. They are represented by rectangles.

Any system can be represented at any level of detail by these four symbols.

Procedure

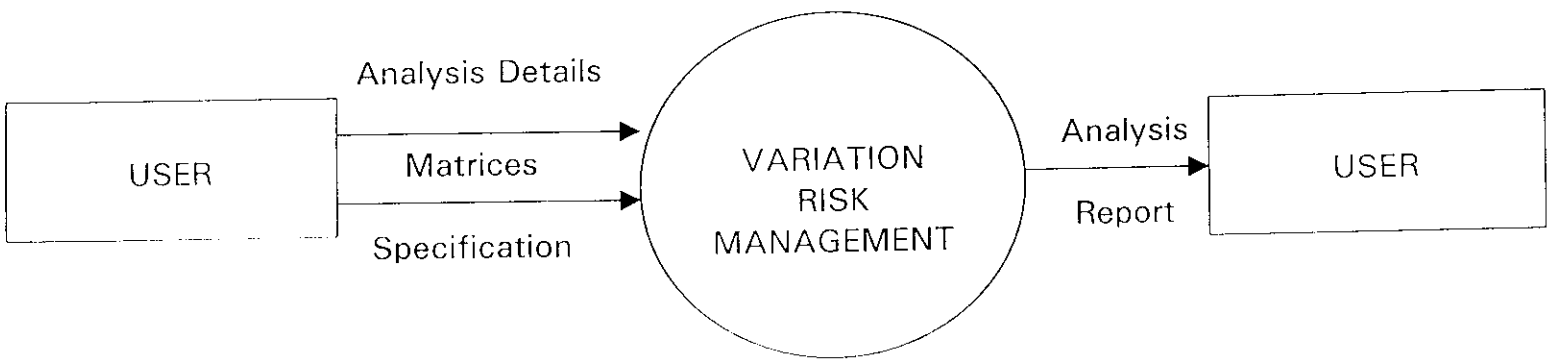
The procedure for producing an architecture flow diagram is to

1. Identify and list external entitles providing inputs / receiving outputs from system;
2. Identify and list inputs from / outputs to external entities;
3. Create a context diagram with system at center and external entities sending and receiving architecture flows;
4. Identify the business functions included within the system boundary;
5. Identify the data connections between business functions;
6. Confirm through personal contact sent data is received and vice-versa.
7. Trace and record what happens to each of the architecture flows entering the system (data movement, data storage, data transformation / processing).
8. Attempt to connect any diagram segments into a rough draft;
9. Verify all architecture flows have a source and destination;
10. Verify data coming out of a data store goes in;
11. Redraw to simplify – ponder and question result;
12. Review with "Informed";
13. Explode and repeat above steps as needed.

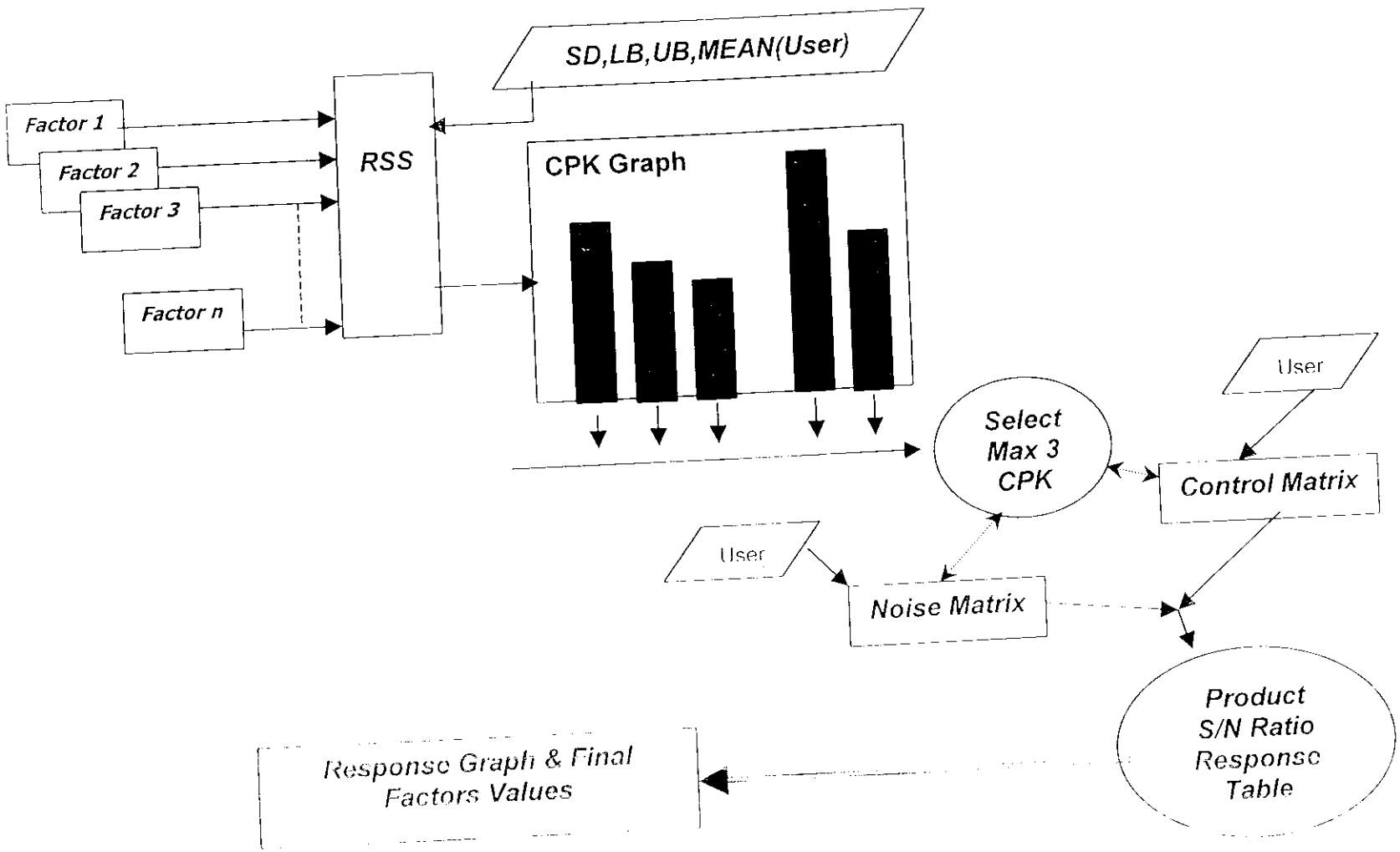
The Architectural Flow Diagram is shown in figure 3.2.1.

Details regarding the process are collected from user. The data are also collected from the user for computation. The data are send to different schemas and process and the final value is produced.

CONTEXT ANALYSIS DIAGRAM



ARCHITECTURAL FLOW DIAGRAM



4. SYSTEM DESIGN

System design is the determination of the overall system architecture, which consists of a set of physical processing components and allocation of system requirements to each component.

The determination of the design and the allocation of requirements is a very interative process. As an iterative component of making technology decision, the functionality expressed by the data flow and control flow diagrams from system requirements analysis is allocated to the various components of the system. The product of the system design is called an architecture model. The model expresses the components of the system, allocation of requirements and topics such as maintenance, reliability, redundancy and self-test. All the requirements are allocated to specific hardware and software configuration items, to manual operations, and so to the interfaces among all system components.

4.1 TABLE DESIGN

DATA FILE

VARIABLE	TYPE	DESCRIPTION
Htype	String	'Horn Type
Atype	String	'Analysis Type
SParam(10)	String	'Selected Factors
MAXCPKText(2)	String	'Maximum Three CPK
TotalParam	Integer	'Total Params
LB(10)	Integer	'LB of Each Param
UB(10)	Integer	'UB of Each Param
Mean(10)	Integer	'Mean of Each Param
SD(10)	Double	'SD of Each Param
MAXCPK(2)	Integer	'MAX 3 CPK Values
CtrlM(3, 2)	Integer	'Control Matrix
NoiseM(2, 3)	Integer	'Noise Matrix
ProdutCmNm(3, 3)	Integer	'CM*NM Product
SnRatio(3)	Integer	'SN Ratio
ResponseTable(5)	Double	'Response Table
RGraph1(1)	Double	'Response Graph 1 Data
RGraph2(1)	Double	'Response Graph 2 Data
RGraph3(1)	Double	'Response Graph 3 Data
O1	Integer	'Optimums
O2	Integer	'Optimums
T1	Integer	'Optimums
T2	Integer	'Optimums
Th1	Integer	'Optimums
Th2	Integer	'Optimums

FACTOR FILE

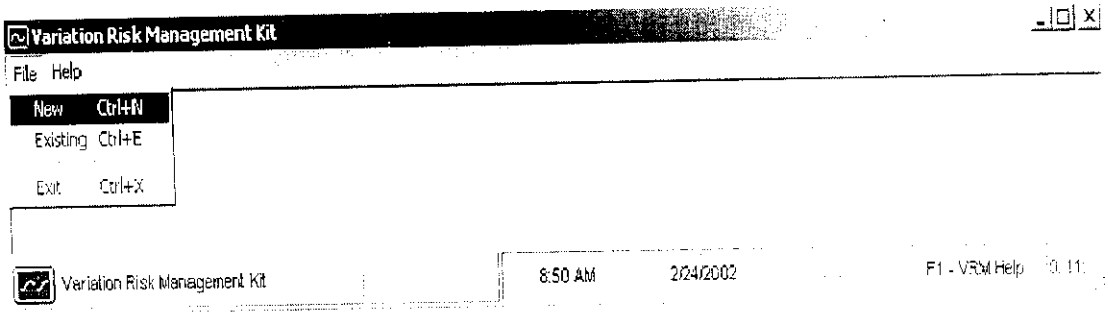
Count	Integer	Number of Factors
Value (200)	String	Factors String Values

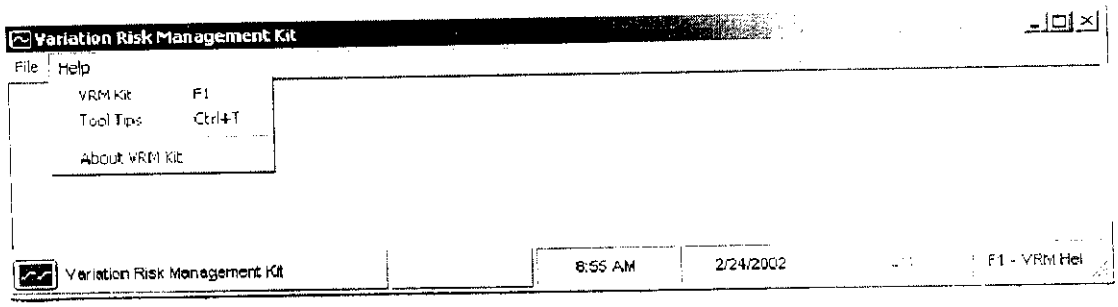
4.2 SCREEN DESIGN

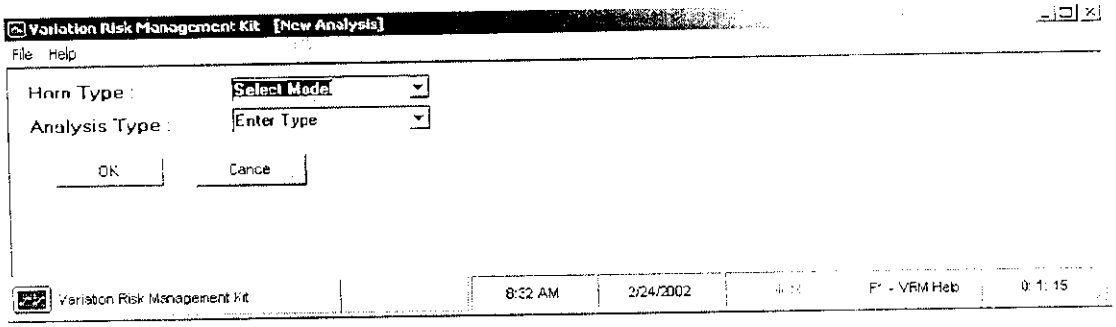
Screen design is the process of covering a user-oriented description of inputs to a computer based programmed oriented specification. The screen design includes the following considerations.

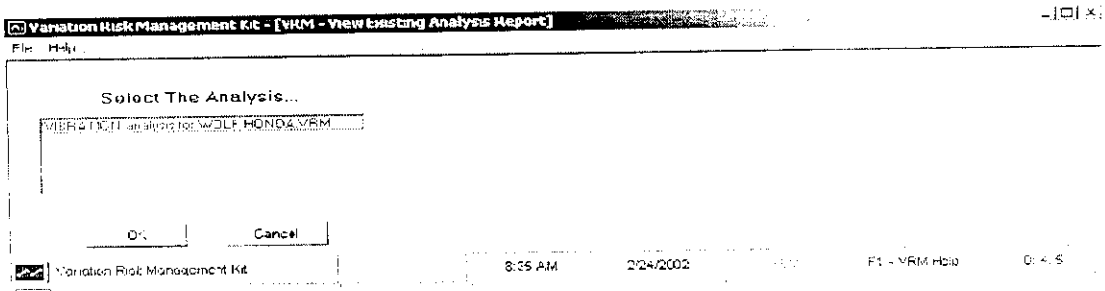
- Collecting only required data and grouping similar or related data.
- The format of the same data in the different screens must be the same.
- On-line help must be provided whenever necessary.
- Consistent terminology must be used throughout the screen.

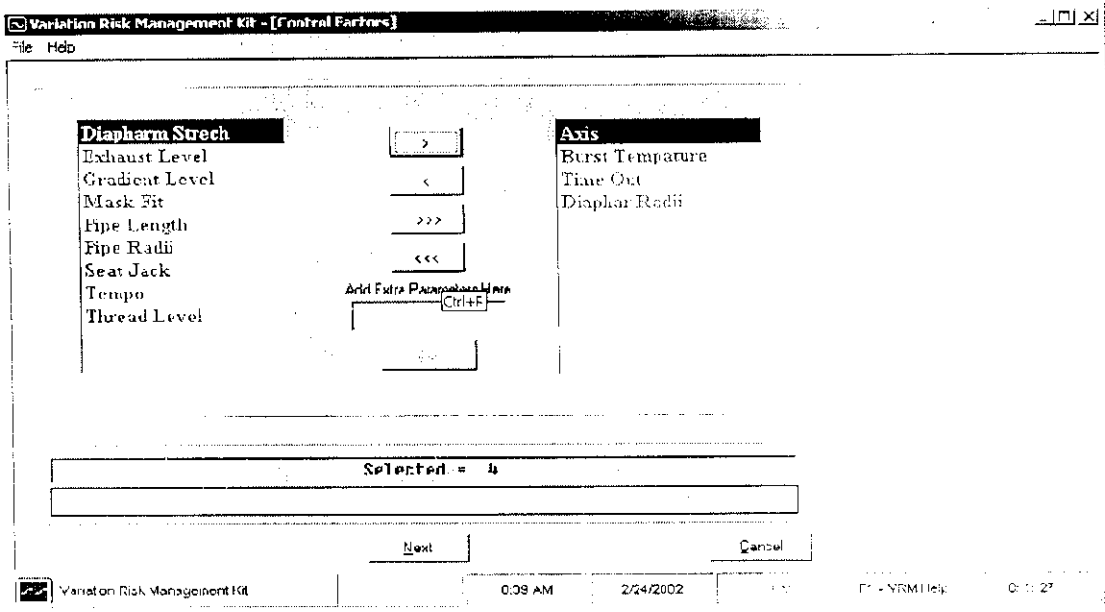
The screen designs are shown from the figures 4.2.1 to 4.2.2.

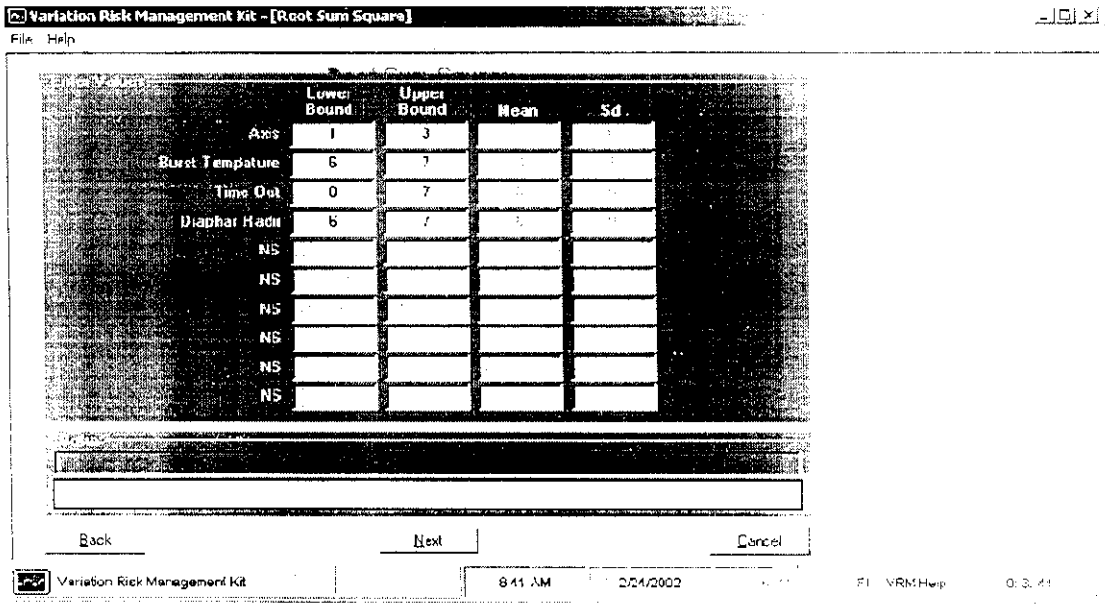

Variation Risk Management Kit
File Help
New Ctrl+N
Existing Ctrl+E
Exit Ctrl+X
Variation Risk Management Kit
8:50 AM
2/24/2002
F1 - VRM Help

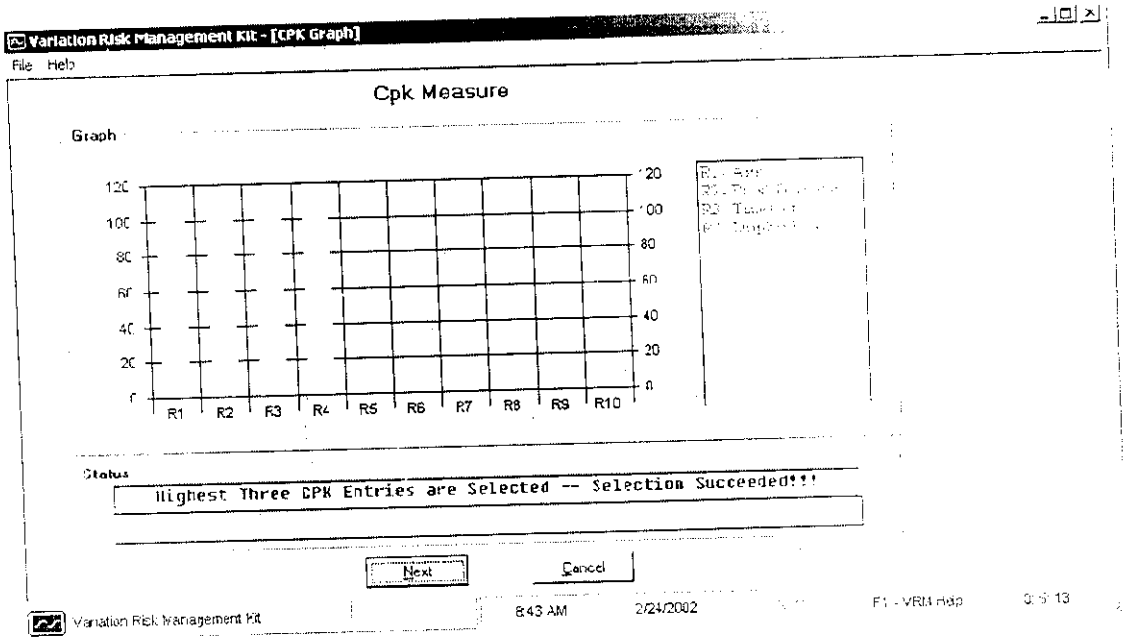

Variation Risk Management Kit
File Help
VRM Kit F1
Tool Tips Ctrl+T
About VRM Kit
Variation Risk Management Kit
8:55 AM
2/24/2002
F1 - VRM Hel

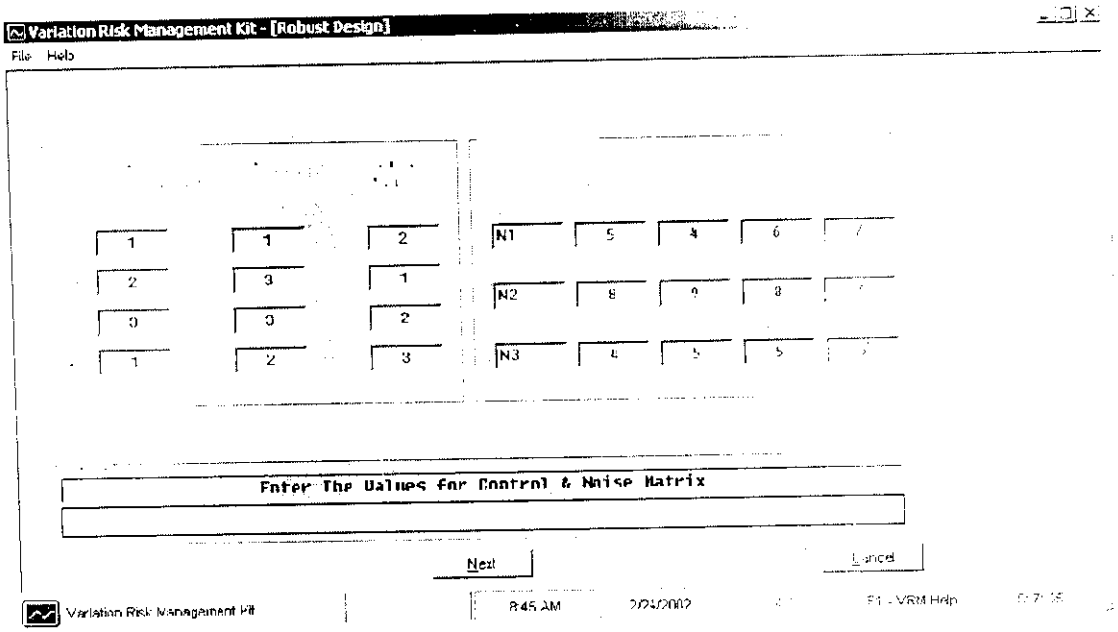

Variation Risk Management Kit [New Analysis]
File Help
Horn Type :
Select Mode
Analysis Type :
Enter Type
OK
Cance
Variation Risk Management Kit
8:32 AM
2/24/2002
F1 - VRM Help

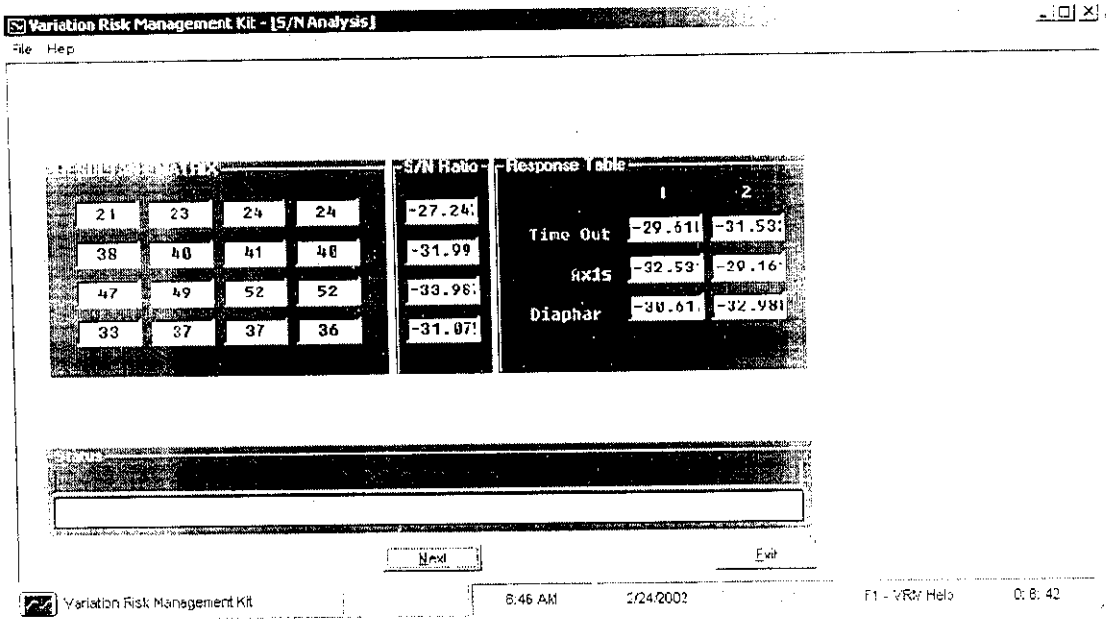

Variation Risk Management Kit - [VRM - View Existing Analysis Report]
Select The Analysis...
OK
Cancel
Variation Risk Management Kit
8:35 AM
2/24/2002
F1 - VRM Help

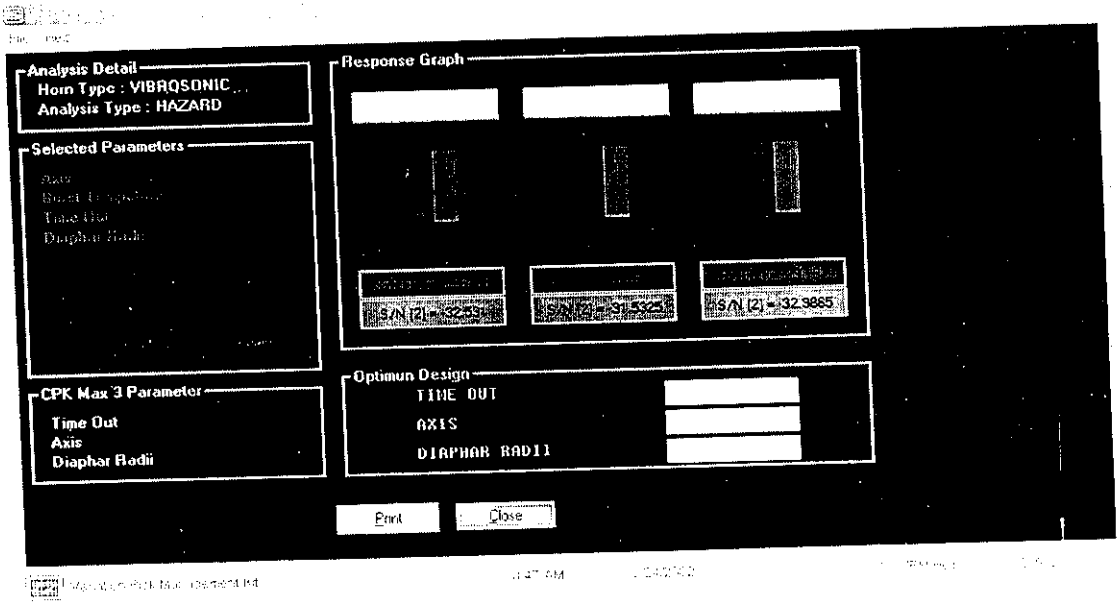

Variation Risk Management Kit - [Control Factors]
File Help
Diapharm Strech
Exhaust Level
Gradient Level
Mask Fit
Pipe Length
Pipe Radii
Seat Jack
Tempo
Thread Level
>
<
>>>
<<<
Add Extra Parameters Here
Ctrl+E
Axis
Burst Temprature
Time Out
Diaphar Radii
Selected = 4
Next
Cancel
Variation Risk Management Kit
0:09 AM
2/24/2002


Variation Risk Management Kit - [Root Sum Square]
File Help
Lower Bound
Upper Bound
Mean
Sd.
Axis
Burst Tempature
Time Out
Diaphar Hadii
NS
Back
Next
Cancel
Variation Risk Management Kit
8:41 AM
2/24/2002
F1 - VRM Help


Variation Risk Management Kit - [CPK Graph]
File Help
Cpk Measure
Graph
120
100
80
60
40
20
0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
Status
Highest Three CPK Entries are Selected -- Selection Succeeded!!!
Next
Cancel
Variation Risk Management Kit
8:43 AM
2/24/2002
F1 - VRM Help


Variation Risk Management Kit - [Robust Design]
File Help
1
1
2
2
3
1
3
3
2
1
2
3
N1
5
4
6
N2
8
N3
4
5
5
Enter The Values for Control & Noise Matrix
Next
Cancel
Variation Risk Management Kit
8:45 AM
2/21/2002
F1 - VRM Help


Variation Risk Management Kit - [S/N Analysis]
File Help
21 23 24 24
38 40 41 40
47 49 52 52
33 37 37 36
S/N Ratio
-27.24
-31.99
-33.96
-31.07
Response Table
1 2
Time Out -29.61 -31.53
Axis -32.53 -29.16
Diaphar -30.61 -32.98
Next
Exit
Variation Risk Management Kit
8:46 AM
2/24/2002
F1 - VRM Help
0: 8: 42


Analysis Detail
Horn Type : VIBROSONIC
Analysis Type : HAZARD
Selected Parameters
CPK Max 3 Parameter
Time Out
Axis
Diaphar Radii
Response Graph
S/N (2) = 32.9885
Optimun Design
TIME OUT
AXIS
DIAPHAR RADII
Print
Close

5. SYSTEM TESTING AND IMPLEMENTATION

System testing is a critical element of software quality assurance and the ultimate review of specification, design and coding. The development of software involves a series of production activities where opportunities for injection of human fallibilities are enormous. Because of human inability to perform and communicate with perfection, software development is accompanied by quality assurance activity.

SOFTWARE TESTING TECHNIQUES

The test case design methods are

i) White Box Testing

ii) Black Box Testing

White Box Testing

Using this testing method it was assured that all the independent paths were exercised at least once. All the logic decisions on their true and false sides were executed. All the loops were executed at their boundaries.

Black Box Testing

Using this technique incorrect and missing functions were rectified and corrected. Interfacing errors, performance errors, initialization errors and termination errors were also found using this technique.

Software Testing Strategies

A strategy for software testing integrates software test case design techniques into well-planned series of steps that results in the successful construction of software. Any testing strategy must incorporate test planning, test case design, test execution and the resultant data collection and evaluation.

Different levels of testing

(i) Unit Testing

(ii) Integrated Testing

(iii) Validation Testing

(iv) System Testing

Unit Testing

The testing of each individual function, module is called unit testing.

Integrated Testing

Integration Testing is a systematic technique for constructing the program structure while at the same time unit tested modules were taken and the program structure that was specified in the design was built and then testing was carried out. The bottom up approach was applied.

Validation Testing

This is carried out to verify whether the software function in a manner that expected by the customer. So ALPHA testing was done to ensure validity.

System Testing

On completion of unit testing for all the units corresponding to every application system undergoes system testing. It is used to test for the functionality of each unit and application as a whole.

Implementation

Once the physical system has been designed in detail, the next stage is to turn design into a working system and then to monitor the operation of the system to ensure that it continues to work efficiently and effectively. Through this it shows that it is ready for implementation.

6. CONCLUSION

The project "Variation Risk Management" Kit has been developed to over come the time consumed processing for the calculation of various risk factors and the preparation of reports whenever required. The system produces all necessary reports for the R & D Department.

6.1 FUTURE ENHANCEMENT

The software has been developed with the present working condition and environments in mind. The current environment is a fast growing area and new features, new technologies and different work styles are expected. Hence this software has been ceveloped with near future needs in mind and it has appropriate slots for any future modifications.

REFERENCES

1. ERIC A. SMITH, "Visual Basic 6.0 Programming Bible"
2. Award Ellias. M, "System Analysis and Design", Sunnel Galgotia Publishing, Second Edition.
3. KENNETH KOZAR, "The Technique of Architecture Flow Diagramming", Oxford University.
4. STEVEN ATTIVA, "Software Design Process", Microsoft Press.
5. HELEN KRISTER, "Visual Basic 6.0 Reference Guide", Microsoft Press.
6. PERVIT MATHEWS, "Encryption Analogy", Kingston Publications.
7. Dr. SHRIRAM, "Environmental Database Architecture", Michigan Institute of Technology.
8. Dr. VIMAL KANNA, "Distributed Application Modelling", University of Ohio.

ONLINE:

www.microsfot.com\msdn

www.howstuffworks.com

http://attila.stevens-tech.edu/~bkhalil/lnotes.html

www.visualbasicforum.com