Unit-1

(1) It is a technique through which we can developed or created Software for comp systems & any other electronic device.

(2) It is a process in which user needs are analized and soft. is designed base on those need.

(3) In S.E the development of S/w Software uses well defined scientific principles, Methods and procedure.

(4) S. Engineers builds soft and app. by using desiging & prog. languages

(5) Inorder to create a complex S/w we should use S/w enering technique as well as to reduce the complexity, we should use abstraction & decomposition

Purpose of S/w E:- 1.) to manage large S/w
2.) for greater scalability
3.) To manage the cost
4.) To manage the dynamic nature of soft ware
5.) for better quality management.

Characteristics of S/w Eng.:-
1.) S/w is developed or engineered, it is not manufactered
2.) S/w does not ware out
3.) S/w is custom built.

Software crisis: It is a term used in early days of comp. sci. for the difficulty of writing useful & efficient prog. in required time

Causes:-

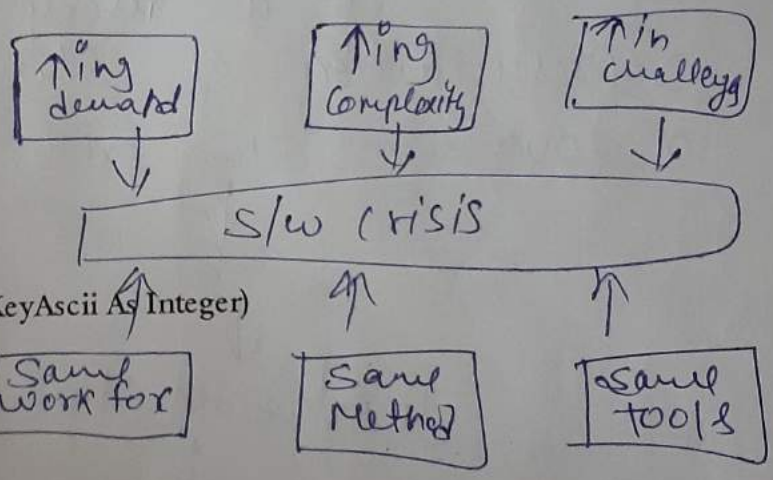
→ project running is over budget

→ project running over time

→ s/w was very Inefficient

→ " " of low quality

→ s/w often did n't meet requirement.



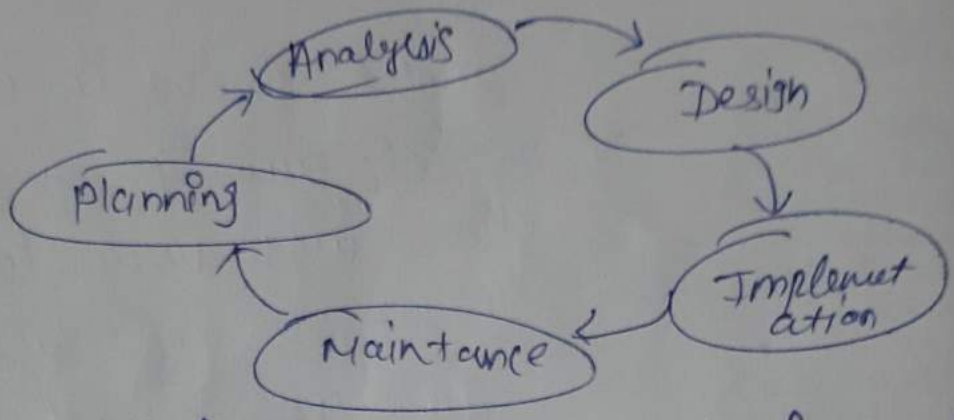
Solution of s/w crisis:-

So E

```
Command2.Enabled = False
End Sub

Public Sub clear()
Text3.Text = " "
Text4.Text = " "
Text5.Text = " "
Text6.Text = " "
End Sub

Public Sub store()
rs(0) = Text1.Text
rs(1) = Text2.Text
rs(2) = Text3.Text
rs(3) = Text4.Text
rs(4) = Text5.Text
rs(5) = Text6.Text
End Sub

Private Sub Label10_Click()

End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If

End Sub

Private Sub Text6_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If
```

Software Develop Life cycle:- OR S/w L C Model OR process

→ key concept of S/w Engineering
→ generally consists of series of stages.



→ S/w L C Model is also called as Process Model.

Definition:- It is a descriptive & diagramatic representation of S/w life cycle

→ To overcome S/w crisis, we use S/w Engineering

↓

S.E

↓

To develop S/w product

↓

Development team must identify suitable life cycle Model for particular project.

→ S/w life cycle Model defines entry & exit criteria for every phase.

→ Without S/w L C M, it's difficult to Monitor progress of project.

A D
P I
M

* Diff. LC Models are! - 1. waterfall →
2. prototype →
3. Spiral →
4. Evaluating →

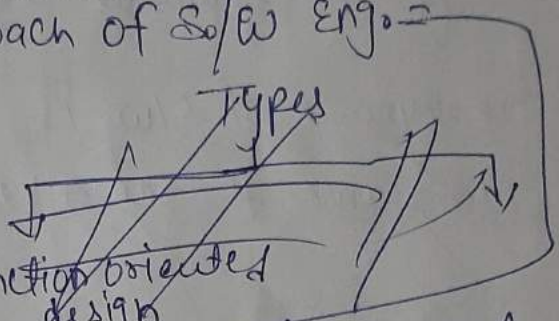
P A D
M I

```
End Sub

Private Sub Text6_LostFocus()
If (val(Text1.Text) + val(Text3.Text)) <val(Text5.Text) Then
  MsgBox "DEFECTS MORE THAN STOCK"
  Text5.Text = " "
  Text6.Text = " "
  Text5.SetFocus
End If
Command1.Enabled = True
Command2.Enabled = True
End Sub

Private Sub Timer1_Timer()
Label18.Caption = Time$
Label2.Caption = Format(Date$, "dd/mm/yyyy")
End Sub
```

Booking:

```
Dim db As Database, rs, rs1 As Recordset

Private Sub Command1_Click()
clear
Command1.Enabled = False
Command4.Enabled = True
Command4.SetFocus
Text1.Visible = False
Text2.Visible = False
End Sub

Private Sub Command2_Click()
Dim FLAG As Integer
FLAG = MsgBox("BOOKING ORDER?", vbOKCancel)
If FLAG = 1 Then
 If rs1.RecordCount <>0 Then
   rs1.Edit
   store
   rs1.Update
   clear
 Else
   rs1.AddNew
   store
   rs1.Update
   clear
```

Approach of S/W Eng.



→ Requirement Analysis & Definition
→ System design
→ program design
→ Writing program
→ Unit testing
→ Integration testing
→ System testing
→ System delivery
→ Maintance.

Challenges of S/w :-

1. problem Domain
2. Soft. is ~~expensive~~ expensive :- Soft-cost 80%. hardware cost 20%.
3. Deployment of S/w is late & unreliable
4. Maintance & reworks.
   - Corrective (debugs, error removing)
   - Adaptive (upgrade)
5. Statisfaction of needs
6. Scale: Size of project ↑ing resources ↑es.
7. Quality & productivity.
8. Consistency & Repeatability.
9. Changes :- Change in busines is rapid, Modification

function point Metrics:- FPM

It overcomes shortcomig of LOC Metric

Why use it? • easily estimate the size of S/w product directly from problem Specification.

• Size of S/w Product & no. of diff. function

1. count the lines
2. LOC
3. func. point i.e. feature point

Steps • When func. is invoked. it reads some input data & transforms it to corresponding output data

• It computes size of S/w product (in unit of function points).

```
 End If
Else
  clear
End If
Command2.Enabled = False
Command1.Enabled = False
Command4.Enabled = True
End Sub

Private Sub Command3_Click()
Unload Me
End Sub

Public Sub clear()
Text1.Text = " "
Label9.Caption = " "
Label10.Caption = " "
Label12.Caption = " "
Label13.Caption = " "
Label16.Caption = " "
Text2.Text = " "
End Sub

Private Sub Command4_Click()
Text1.Visible = True
clear
Text1.SetFocus
Command4.Enabled = False
Command1.Enabled = True
End Sub

Private Sub Form_Load()
Set db = OpenDatabase("c:\program files\microsoft visual
studio\Vb98\gasagency.mdb")
Set rs = db.OpenRecordset("CUSTMAST")
Set rs1 = db.OpenRecordset("TRANSACTION")
Command2.Enabled = False
Command1.Enabled = False
Text1.Visible = False
Text2.Visible = False
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
  SendKeys "{TAB}"
End If
```

* func. point computed in two steps:-

1) UFP:- (Unadjusted FP)

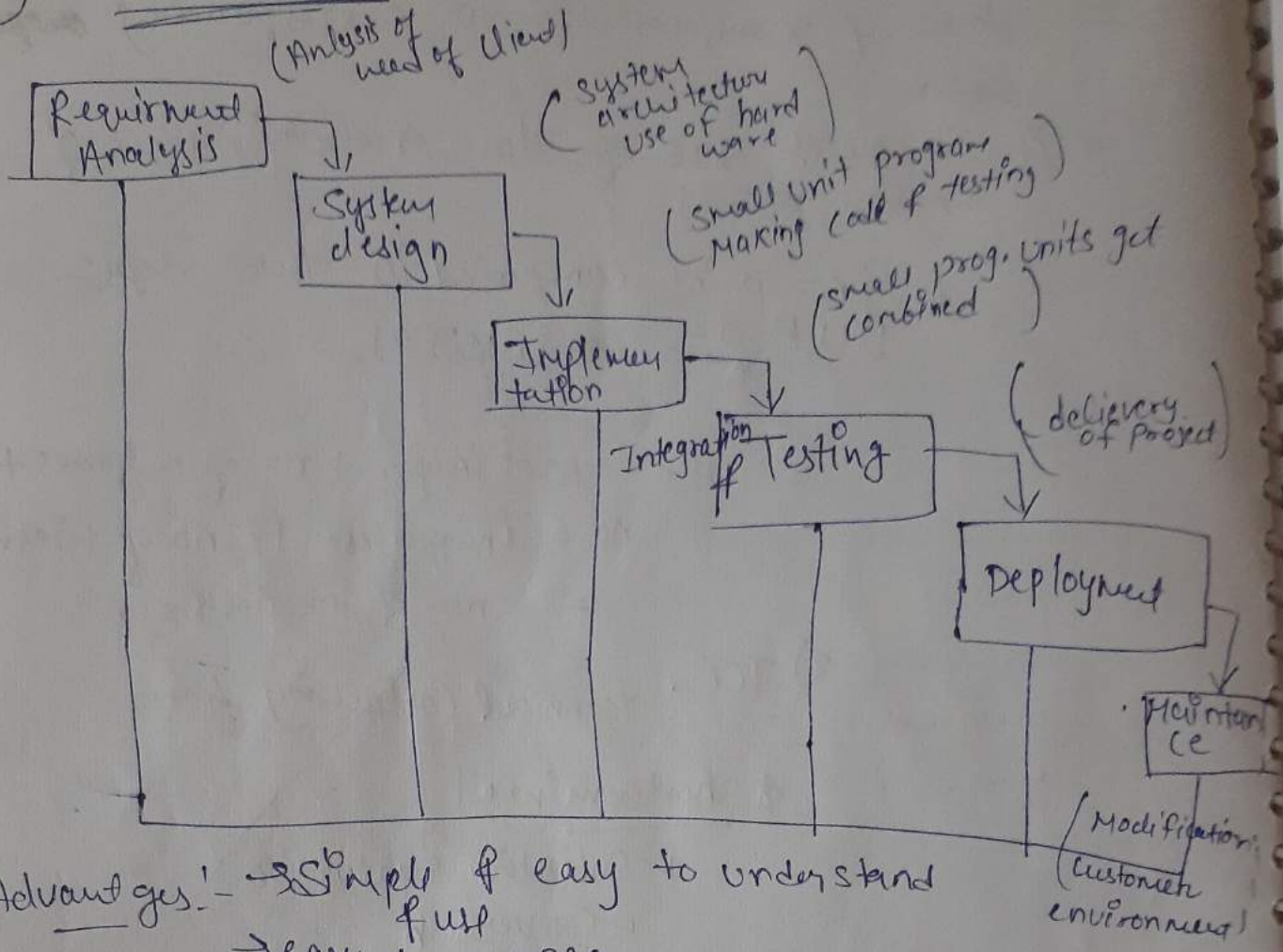
UFP = no. of inputs*4 + no. of outputs*5 + no. of inquiries*4 + no. of files*10 + no. of interfaces*10

2) TCF = technical complexity factor

* Shortcoming:-

- Subjective evaluation
- Conversion
- Less research data
- Late Performance
- Low Accuracy
- Long learning curve

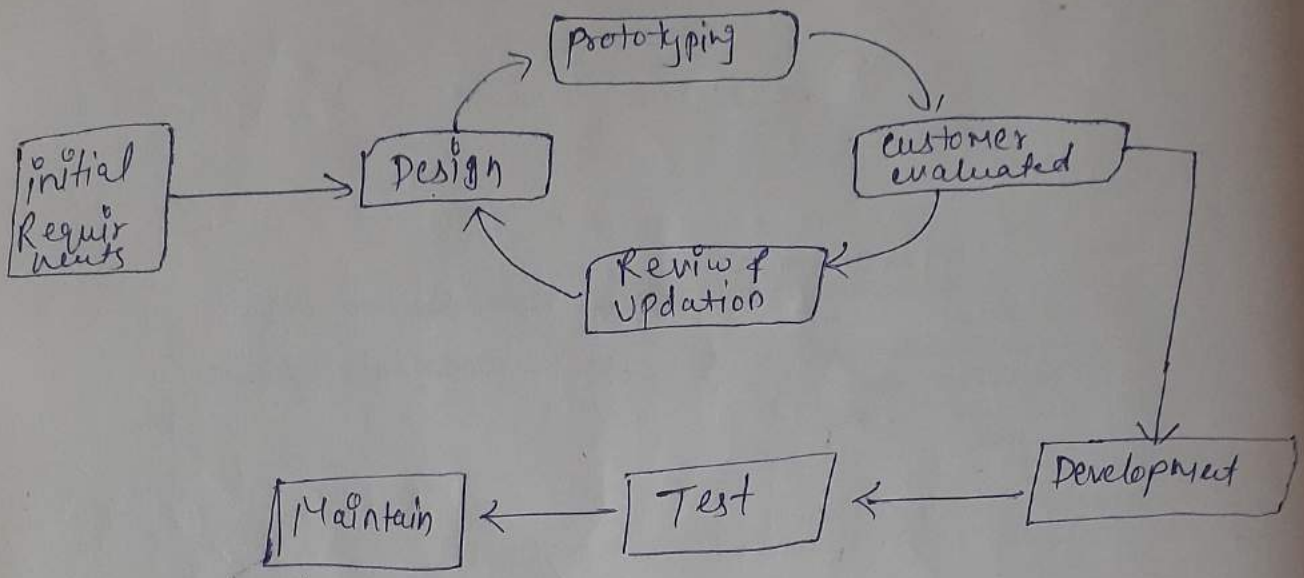
(1) Waterfall Model:-



Advantages:- → Simple & easy to understand & use
→ easy to manage
→ phase are proceded & complete one at a time
→ requirements are very well understand
→ Clearly defined stages

Disadvantages → no working is produce untill late during life cycle
→ high amount of risk
→ not good for complex or object orientes program
→ Poor Model for large & ongoing projects

(2) Prototyping Model:



- Advantages:-
  1) Customers can see steady progress.
  2) Useful when Requirement are ~~chang~~ changing rapidly.
  3)

Disadvantage:- 1.) Impossible to know how long it'll take.
2.) There's no way to know, no. of iterations requirement.

(3) Time-Boxing Model:-

→ development is done iteratively as in iterative enchanment model.
→ Each iteration is done in a time box of fixed duration.
→ functionality to be developed is adjusted to fit the duration fit the duration of the time box.
→ Each timebox divided into a sequence of fixed stages.
→ each stages performs clearly defined tasks. (Analysis, Implementation & Deployment)

→ Time duration of each stage is approx. equal.

→ Stages ~~in which~~ should be chosen in such a way that each stage perform some logical unit of work that becomes input of next stage.

```
    Text2.Visible = False
  End If
Else
    Label9.Caption = rs(1)
    Label10.Caption = rs(2)
    Label12.Caption = rs(3)
    Label13.Caption = rs(4)
    Label16.Caption = rs(6)
    Text2.Visible = True
    Text2.SetFocus
End If
ABC:
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If
End Sub

Private Sub Text2_LostFocus()
```

TB1 TB2 TB3

Analysis → Implem

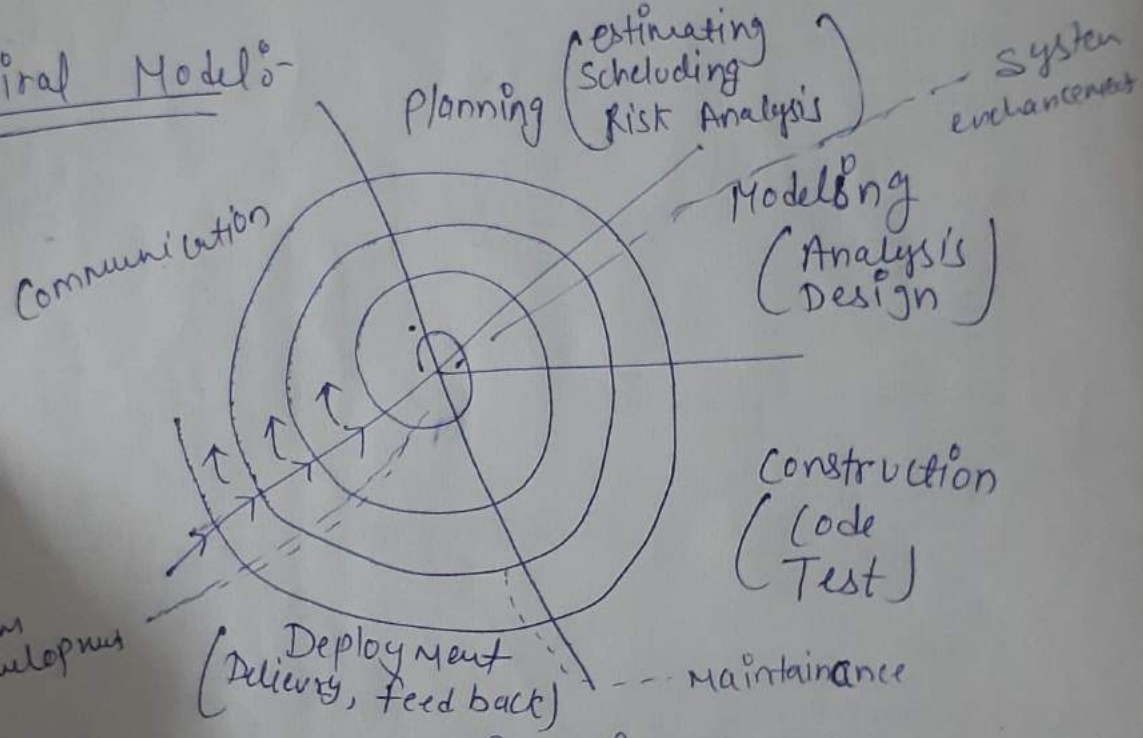
Analysis → Impl

Analysis

Advantages:- → Speeds up development process & shorten delievery time.
→ Suited to develop projects with a no. of features in short time period.

Disadvantages:- → Project Managment becomes more complex.
→ not good for project in which work can't be divided into iteration of equal duration.

4. Spiral Model:-



* Combination of waterfall & Iterative Model
* each phase in spiral Model begins with design goal & end with client reviewing
* S/w developed in a series of incremental releases

*When to use Spiral Model:-
→ when project is large
→ releases are required to be frequent.
→ risk & cost evaluation is important
→ requirments are unclear & complex.

→ change may require at any time
→ for medium to high risk object.

* Advantages:-

→ Additional functionality or change can be done at later stage

→ Cost estimation becomes easy

→ develop is fast & systematic

→ space for client's feedback.

* Disadvantages:-

→ Risk is not meeting the schedule or budget

→ Documentation is more

→ not good for smaller project, it'll cost a lot.

```
End If
If Label16.Caption = "d" Then
 If val(Text2.Text) >2 Then
    MsgBox "MORE THAN 2 CYLINDER NOT POSSIBLE"
    Text2.Text = " "
    Text2.SetFocus
    Command2.Enabled = False
    FLAG = True
 End If
End If
If FLAG = False Then
  Command2.Enabled = True
End If
ABC:
End Sub

Public Sub store()
rs1(0) = val(Text1.Text)
rs1(1) = rs(6)
rs1(2) = Date$
rs1(3) = "n"
rs1(4) = DateAdd("d", 15, rs1(2))
rs1(5) = val(Text2.Text)
End Sub
```

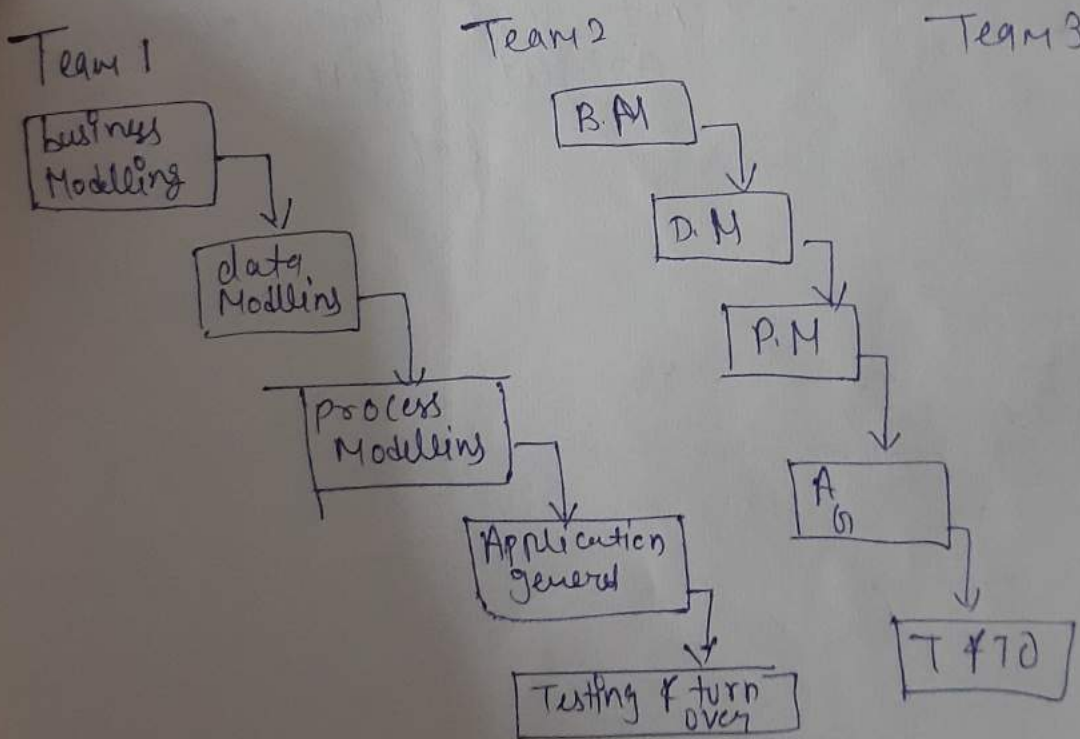
Delivery:

```
Dim db As Database, rs, rs1 As Recordset
Dim val As Integer

Private Sub ext_Click()
MDIForm1.Enabled = True
Unload Me
End Sub

Private Sub Form_Load()
Set db = OpenDatabase("c:\program files\microsoft visual
studio\Vb98\gasagency.mdb")
Set rs = db.OpenRecordset("TRANSACTION")
If rs.RecordCount <>0 Then
 On Error GoTo a
 rs.MoveFirst
 Do While Not (rs.EOF = True)
  If (DateDiff("d", Date, rs(4)) <= 0) Then
   If rs(3) = "n" Then
```

(5.) RAD Model:- Rapid Application Development

→ It is an Increment Model that emphasies on extremly short development cycle. (60 to 90 days)

→ phases:- (1.) Communication
(2.) Planning
(3.) Modeling Includes → business Modelling
→ Data Modelling
→ process Modelling
(4.) Construction
(5.) Deployment.



→ It uses components based construction
→ uses multiple team on scable project
→ requires heavy resource
→ required developers, & customer who are heavily committed.

```
    Combo1.AddItem rs(0)
    rs.MoveNext
   Else
    GoTo c
   End If
  Else
c:
    rs.MoveNext
  End If
 Loop
End If
a:
End Sub

Private Sub prn_Click()
Form6.Show
End Sub

Private Sub Timer1_Timer()
Label5.Caption = Time$
Label6.Caption = Date$
End Sub
```

→ Performance can be a problem
→ difficult to use with new technologies.

#. Automation through S/W Environment

It generally refers to use of digital technology simply to work & perform process.

Billing:

```
Dim db As Database, rs, rs1, rs2, rs3, rs4 As Recordset
Dim k As Integer

Private Sub Command1_Click()
Dim FLAG As Integer
rs4.Edit
If Label14.Caption = "c" Then
  If rs4(0) <20 Then
    FLAG = MsgBox("RE ORDER COMMERCIAL STOCK,WANT TO ADD
STOCK?", vbYesNo)
    If FLAG = 6 Then
      Form3.Show
     ElseIf FLAG = 7 Then
        GoTo a
     End If
   Else
     rs4(0) = rs4(0) - val(Label16.Caption)
   End If
  Else
```

#. Quality Std:- "A quality product does exactly what the users want it to do".

* S/w product, several quality factors:-
  1. portability ⇒ platform Independent
  2. Usability
  3. Reusability
  4. Correctance
  5. Maintainance

#. (1.) SEI-CMM:- (Quality std Inst.) developed
Software Enginring Institution ⇒ Capability Maturity Model

→ Helped organization to improve quality of S/w they developed
→ Adoption of SEI-CMM has significant busniess benefits
→ SEI-CMM can be used in 2 ways

Capability evalution
⇓
provide a way to access the S/w process capability in a organization.

Software Process Assessment
⇓
It's used by organization with the objective to prove process capability

→ SEI-CMM S/w developed Industries into 5 Maturity levels

Level 1 initial level → A s/w develo. organization at this level is charactized by adhoc activities.

Level 2 Repeatable → At this level Basic project maganet practices such as tracking cost, f schedule are established

Level 3 Defined → At this level, process for both managnet f' develonent activities are defined f documented.

Level 4 Managed → At this level, focus is on s/w Matrices

Level 5 Optimizing → At this stage processes f product Matrices are collected.

```
If rs4(0) <20 Then
  FLAG = MsgBox("RE ORDER DOMESTIC STOCK,WANT TO ADD
STOCK?", vbYesNo)
  If FLAG = 6 Then
    Form3.Show
  ElseIf FLAG = 7 Then
    GoTo a
  End If
 Else
  rs4(1) = rs4(1) - val(Label16.Caption)
 End If
End If
rs4.Update
k = Form5.Combo1.ListIndex
Form5.Combo1.RemoveItem (k)
rs3.AddNew
rs3(0) = Label19.Caption
rs3(1) = Label11.Caption
rs3(2) = Label2.Caption
rs3.Update
Set rs1 = db.OpenRecordset("select * from TRANSACTION where cons_no="
& val(Label11.Caption))
rs1.Edit
rs1(3) = "y"
rs1(4) = Date$
rs1.Update
MDIForm1.Enabled = True
a:
 Unload Me
 Form5.Visible = False
End Sub

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Form_Load()
Set db = OpenDatabase("c:\program files\microsoft visual
studio\Vb98\gasagency.mdb")
Set rs2 = db.OpenRecordset("PRICE")
Set rs3 = db.OpenRecordset("BILL")
Set rs4 = db.OpenRecordset("STOCK")
If rs3.RecordCount = 0 Then
  Label19.Caption = 1000
Else
  rs3.MoveLast
```

(20) ISO 9001 Std:- (International organization for Standardization)

* ISO 9001 applies to organization which are involved in design, development, production & servicing of goods
* This Std is applicable to most S/W development organization.
* Features of ISO 9001 certifications:-

→ All doc concerned with the development of S/W product should properly Managed, authorized & controlled.
→ Imp. doc. should be independantly checked & reviewed for correctness &
→ Product should be checked or tested against specifications.
→ Several organizational aspects should be addressed eg:- Management reporting of Quality team
→ It use TQM technology

ISO 9000	CMM
1.) Std. developed by ISO	1.) It's a Model developed by SEI
2.) applicable for any type of Industry	2.) Applical for Soft. Industry
3.) verify quality of product & provide certificate to orga.	3.) verifies Quality of S/W, acc^n to Quality provide Level.

4.) no level
5) 100+ countries accepted
6) either pass or fail

4.) has 5 level
5) Mostly used in USA
6.) provide Maturity level.

```
    Label19.Caption = rs3(0) + 1
  End If
  k = Form5.Combo1.Text
  Set rs = db.OpenRecordset(" select * from CUSTMAST where Cons_no=" & k)
  Set rs1 = db.OpenRecordset(" select * from TRANSACTION where cons_no="
  & k)
  If rs1(3) = " n" Then
    load
  Else
    Command1.Enabled = False
  End If
  End Sub

  Private Sub Timer1_Timer()
  Label5.Caption = Time$
  Label2.Caption = Date$
  End Sub

  Public Sub load()
  Label11.Caption = rs(0)
  Label12.Caption = rs(1)
  Label13.Caption = rs(2)
  Label14.Caption = rs(6)
  Label16.Caption = rs1(5)
  If rs1(1) = " d" Then
    rs2.MoveFirst
    Label17.Caption = rs1(5) * rs2(1)
  Else
    rs2.MoveLast
    Label17.Caption = rs1(5) * rs2(1)
  End If
  End Sub
```

Returns:

```
Dim db As Database, rs As Recordset

Private Sub Command1_Click()
rs.Edit
rs(5) = rs(5) - val(Text1.Text)
rs(4) = rs(4) - val(Text2.Text)
rs.Update
MsgBox " GOODS RETURNED"
```

Software Requirement Specifications:- (SRS) documentation.

A requirement specification for a S/w system is a complete description of the behaviour of system to be developed.

→ It includes set of use cases, that describes all interaction, the users will have with system

→ SRS consists of Functional Requirements:-
  ⇒ describes functionality of system services
  ⇒ depends on type of S/w, & users

→ Non-functional Requirements:-
  ⇒ defines system's properties & constraints
  eg:- Reliability, Respond time
  Storage requirement.

SRS

★ Components:-

1. Introduction:-
   - purpose
   - scope
   - definition
   - System overview
   - Reference

2. Overall Description:-
   - product perspective
   - product functions
   - User characteristics
   - Constraints, Assumptions & Dependencies

3. Specific Requirement:-

- External interface
- functions
- Performance Requirement
- logical db Require.
- Design constraint
- key features

4. Validation checks:-

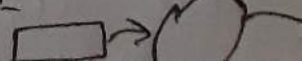
- Validity check
- Consistency check
- Realism check
- Verifiability check
- Completness check

5. Validation Techniques:-

- Requirment reviews
- prototyping techniques
- Test case generation

SRS includes behavioral & non-behavioral requirments

Level of DFD:- 1. 0th level:-



```
Label6.Caption = rs(4)
Label5.Caption = rs(5)
End Sub

Private Sub Command2_Click()
clear
End Sub

Private Sub Command3_Click()
Unload Me
End Sub

Private Sub Form_Load()
Set db = OpenDatabase("c:\program files\microsoft visual
studio\Vb98\gasagency.mdb")
Set rs = db.OpenRecordset("STOCK")
Label6.Caption = rs(4)
Label5.Caption = rs(5)
End Sub

Public Sub clear()
Text1.Text = " "
Text1.SetFocus
Text2.Text = " "
Label5.Caption = " "
Label6.Caption = " "
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If

End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
```

#. SRS includes:-
→ Behavioral Requisents
→ Non Behavioral Requirments

#(1.) Behavioral Requirmts:- ⇒ Defines "what the system does"
- i/p
- o/p
- transformation of i/p to o/p

(2.) Non-Behavioral Req:- It defines "attributes of the system as it performs it's job"
- efficiency
- Reliability
- Security
- Maintainability
- portability
- Std Compliant

Requirement Analysis:- It is the process of defining, documenting & maintaing the requirments. It is a process of gathering ~~consists~~ and defining, services provided by the system Requirmt Enguring process
- Data flow diagram aka 'Bubble chart',

Consists of following main activities

* Requirment Elicitation
* " Specification
* " Verification & Validation
* " Managment
* " Analysis & Negotiation

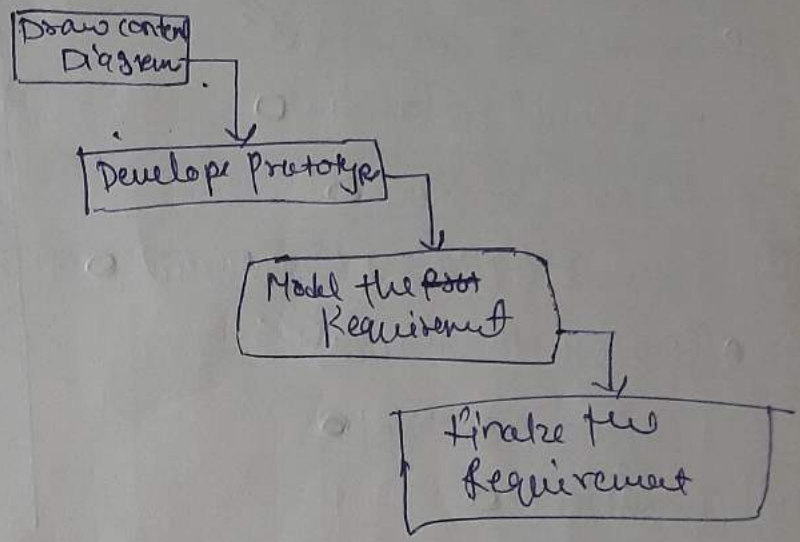
Daily report:

```
Private Sub Command1_Click()
DataReport3.Show
End Sub

Private Sub Command2_Click()
DataReport4.Show
End Sub

Private Sub Command3_Click()
Unload Me
End Sub
```

Datewise report:

```
Dim a As Integer
Private Sub Command1_Click()
If IsDate(Text1.Text) = True Then
  DataEnvironment1.rsCommand5.Open ("select * from transaction where
day(released_date)=" & day(Text1.Text) & " and month(released_date)=" &
Month(Text1.Text)), DataEnvironment1.Connection1, adOpenKeyset,
adLockOptimistic
  DataReport5.Show
Else
  MsgBox "ENTER VALID DATE"
  Text1.Text = ""
  Text1.SetFocus
End If

End Sub
Private Sub Command2_Click()
Unload Me
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If
End Sub
```

Req. Analysis steps:-



#Structured Analysis:-

1) It's a systematic development method that allow us to understand overall functionality of system and it's diff. activities in a graphical representation [DFD]

2.) decomposed high level func. into no. of function

3.) It has 3-measure Principle
- ⊙ Top-down decomposition
- ⊙ Devide & conquer
- ⊙ graphical representation of Analyst result

Rule:-
1. atleast one i/p & one o/p
2. atleast one inflow & one flowout.
3. either data get store or process in
4.

~~consists~~ and defining, services provided by the system Requirment Engeering process

DfD = • Data flow diagram aka 'Bubble chart',
• graphical representation of data
• easy understanding of functionality
• tells how data processed

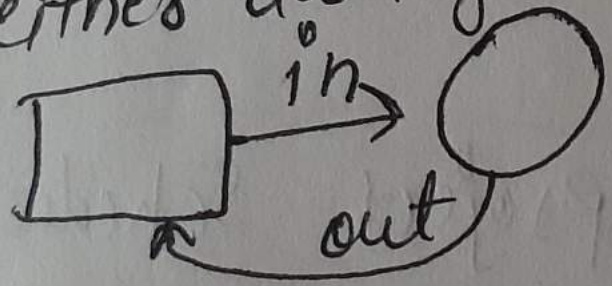
• Notations:- Ext. Entity → □ →
Data flow →

Process or bubble ○
Data Store ═══

eg- 'LED
• display:- Hard

```
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If
End Sub
```

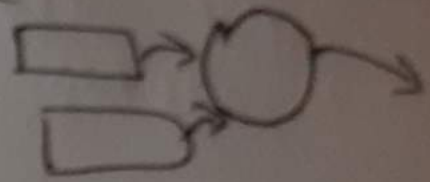
Rule :-
1. atleast one i/p & one o/p
2. atleast one inflow & one flowOut.
3. either data get store or process
4.



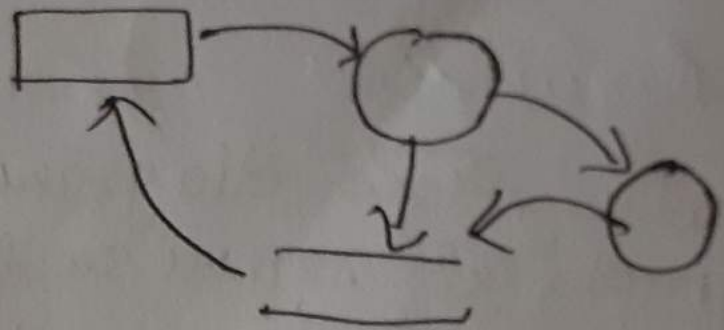
```
End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
SendKeys "{TAB}"
End If
End Sub
```

non behavioral requirments

1. Level of DFD:- 0th level:-



2. 1st level:-



3. 2nd level:-

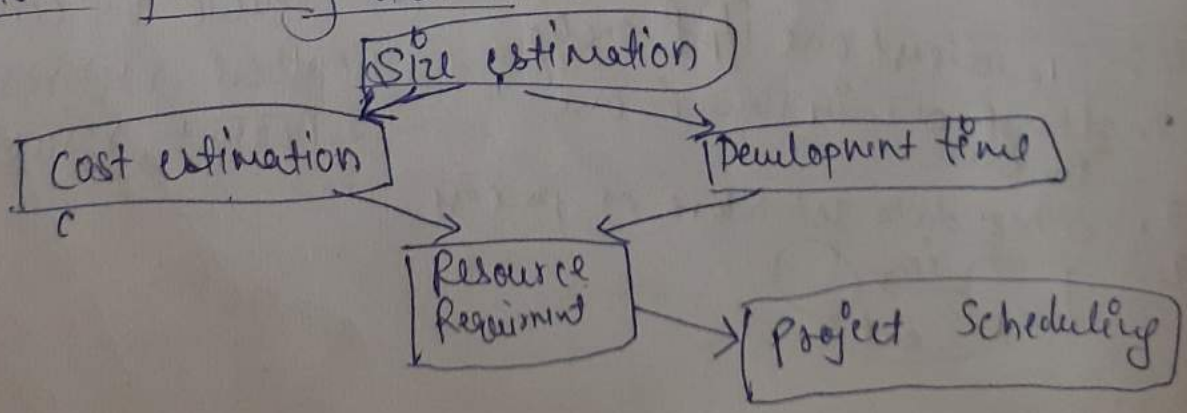
3. COCOMO Model
4. Putnam Resource Allocation Model
5. Risk Management
6. project scheduling
7. personnel planning
8. team structure
9. Soft. configuration Management
10. Quality Assurance
11. project Monitoring
12. Empirical.

4. function oriented design
5. Cohension, coupling & their classification
6. User Interface Design
7. Detailed design
8. Information flow Metric

#. Sw Project planning :- In order to conduct a Sucessfull sw project, we must understand the

→ risk to be incurred.
→ resource requirement
→ Scope of work to be done
→ task to be accomplished
→ cost to be expended
→ Schedule to be followed

* Various planning activities :-

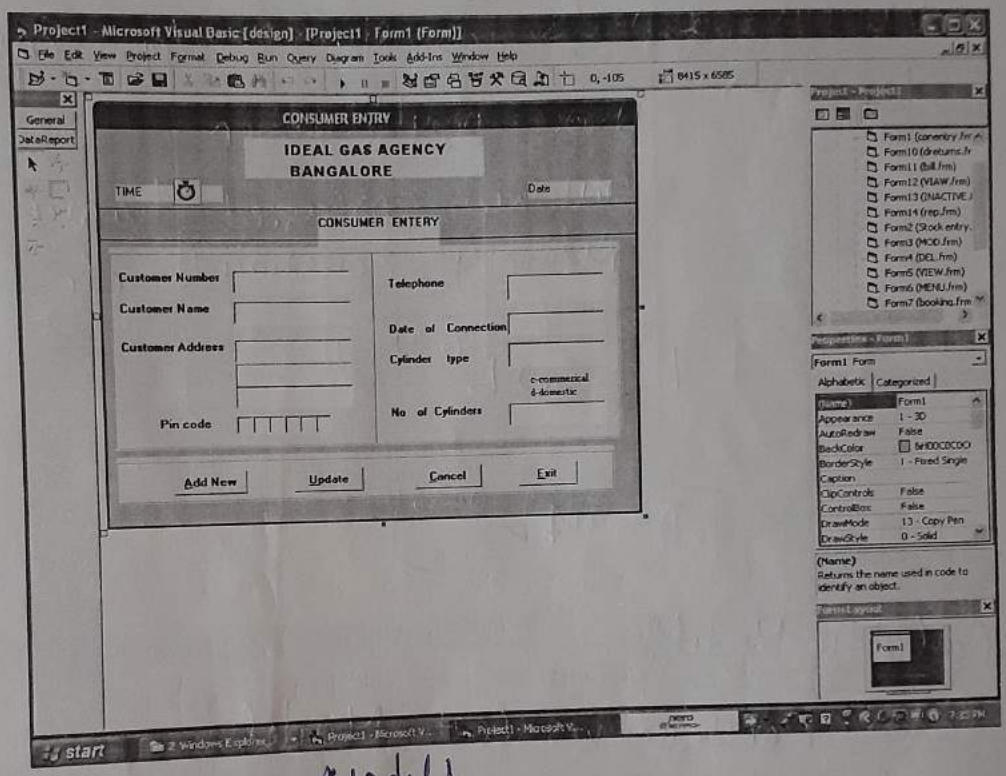


* Non-Emissive display :- It use optical

(1.) Cost Estimation:- No. of estimation techniques have been developed & having some attributes in common -

* Project Scope must be established in advanced
* S/w Metrice are used as basic from which estimates are made
* Project is divided into small pieces which're estimated individually.

d). SCREENS:



Models

Static, Single Variable Model

Static multivariable Model

* Static Single Variable Model:- (single) → (size)

In static single Model a unique variable taken as unique ~~variable~~ element for calculating all other factors (like (time, cost).

$C = aL^b$ → size of code

a, b = constant

$E = 1.4L^{0.93}$

Cost effort

$DOC = 30.4L^{0.90}$ documentation in no. of pages

$D = 4.6L^{0.26}$ ⇒ duration of time

* Static Multivariable Model :-

These Models are often based on equation $C = aL^b$ they actually depend on several variables, representing various aspects of s/w Develop. Environment for eg → user participation
Customer oriented changes
Memory constraints etc.

$E = 5.2L^{0.91}$

$P = 4.1L^{0.36}$

#. CoCoMo Model :-

Constructive Cost Model

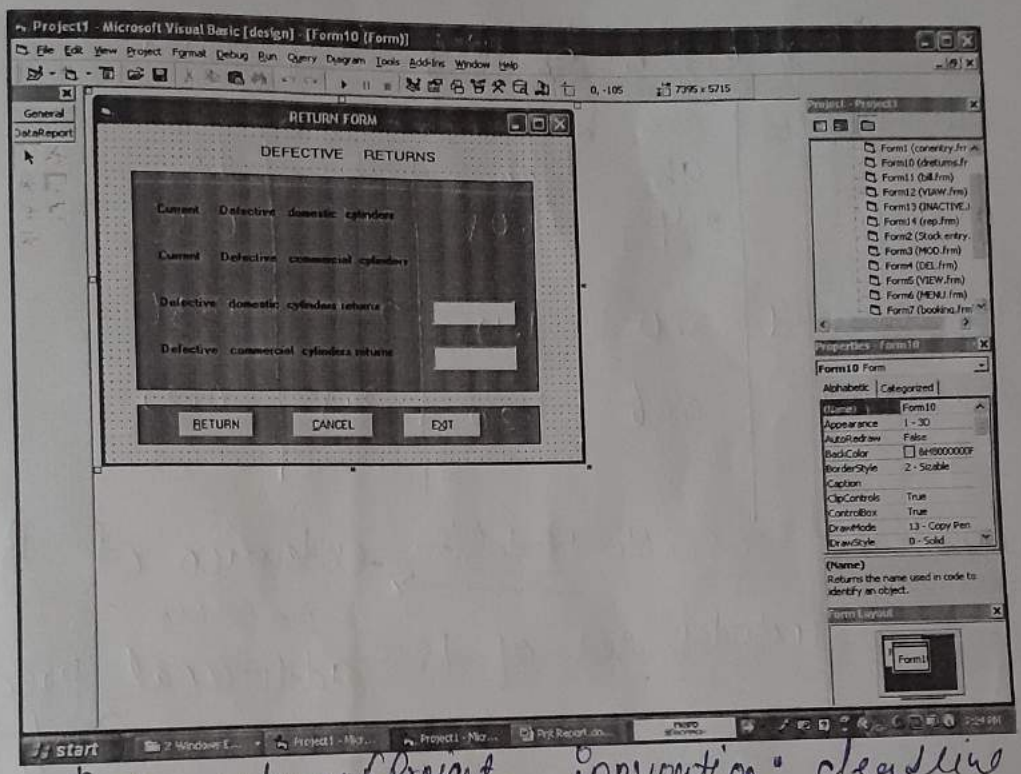
⇒ used for :- • calculate the effort, that a person can do
• development time
• Average staff size
• Productive

⇒ COCOMO include :- * Basic COCOMO Mode
* Intermediate COCOMO Model
* Complete or detailed COCOMO

⇒ COCOMO applied on 3 classes of SW Modes:

1. organic Mode
2. Semi-detached Mode
3. Embedded Mode

(1.) Organic Mode:- project size, nature of project, innovation, deadline of project.



project size; nature of project, innovation; deadline of project
2-50K LOC; small size; little; no tight schedule.
eg- Pay Roll

(2.) Semi-detached Mode:-

50-300K loc; Medium size project; Medium; Medium
Medium size team
eg- dbsys.

(3.) Embedded:-

Over 300K loc; large project; significant; tight
real time sy
complex interface
eg:- ATMs

* Basic CoCoMo Models :-

$E = a_b(KLOC)^{b_b}$ (effort applied)

$D = c_b(E)^{d_b}$ (Development time)

People required $P = E/D$

a_b, c_b, d_b, b_b are coefficient.

S/w project	can be work on a_b	b_b	c_b	d_b
organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

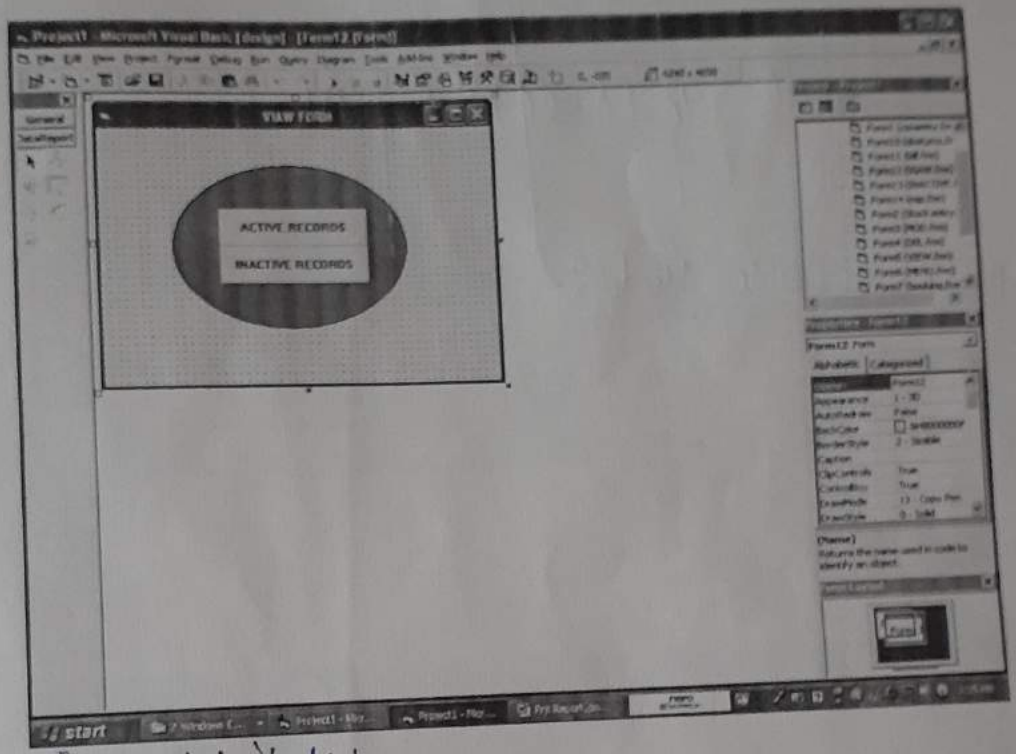
* Intermediate CoCoMo Model :- → extension of Basic CoCoMo
→ includes set of 15 additional predictors (cost drivers)
// cost driver adjust nominal cost of project to actual proj. environment.

15 cost Drivers :-

(1) Product attributes :- - require s/w Relia (RELY)
- dB Size (DATA)
- product Complexity (CPLX)

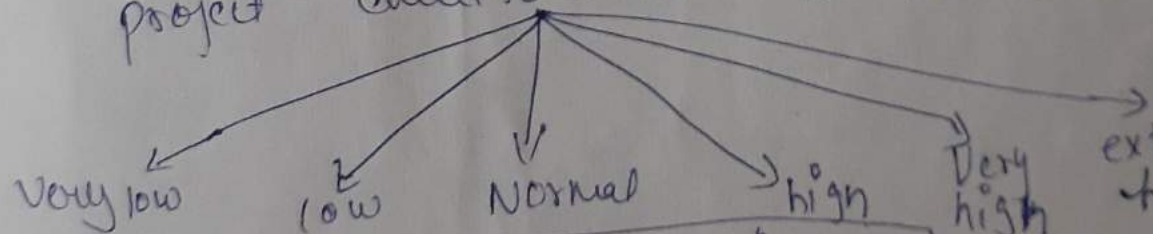
(2) Computer attributes :-
- execution time constraints (TIME)
- Main Storage Cons. (STOR)
- Virtual Machine Volality (VIRT)
- Computer turn around time (TURN)

(3) Personal attributes:-
- Analyst capability (ACAP)
- App. Experience (AEXP)
- programmer capability (PCAP)
- Virtual Machine Experi. (VEXP)
- prog. langug Experi (LEXP)



(4) Project Attributes:-
- Modern prog. practices (MODP
- Use of soft tools (TOOL)
- required develop schedul) (SCED

→ each cost drivers are created for given project environment in terms of-



$$\text{Eqn} = E = a_i^b (KLOC)^{b_i^b} \times EAf$$
$$D = C_i^b (E_i^b)^{d_i^b}$$

(effort Adjustment factor)

project	a_i	b_i	c_i	d_i
Organic	3.2	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
embedded	2.8	1.20	2.5	0.32

$$TE = a_i (KLOC)^{b_i} * EAf$$

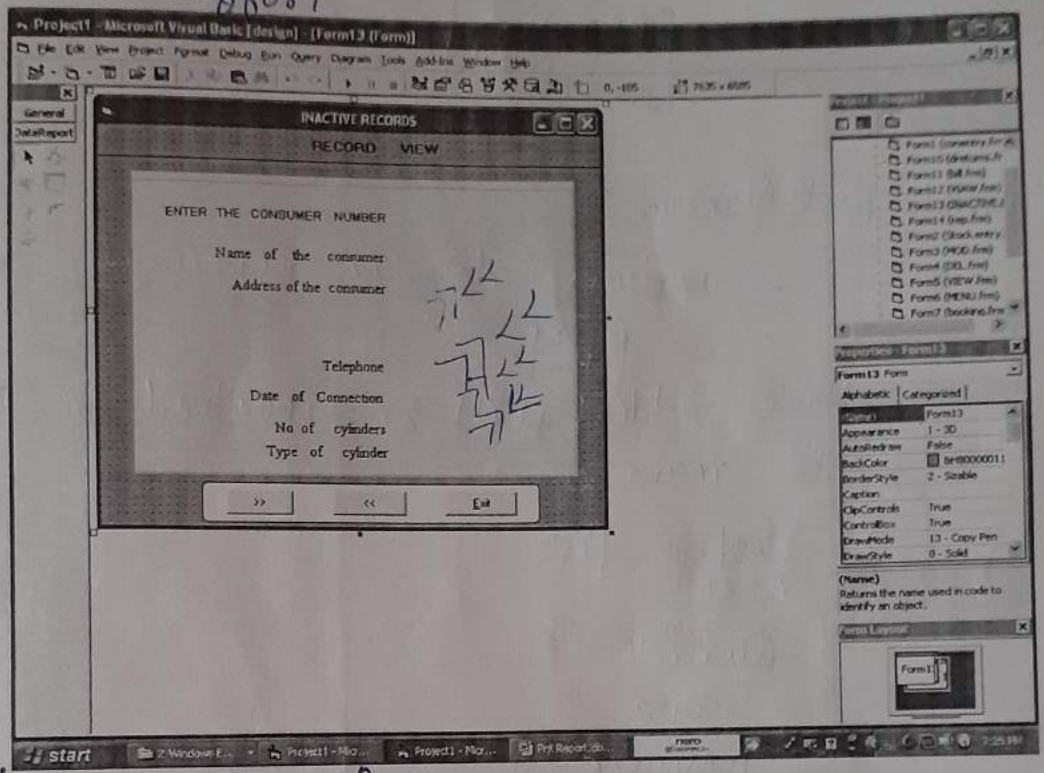
* Complete COCOMO Model:-

Putnam Resource Allocation Model:-

$$L = C_k K^{1/3} t_d^{4/3}$$

C_k : state of technology const → 2000 poor S/w, 8000 good S/w Environ, 11000 excellent

K : Deliverd LOC effort



t_d = Develop time

$$K = \frac{L^3}{C_k^3 \, t_d^4}$$

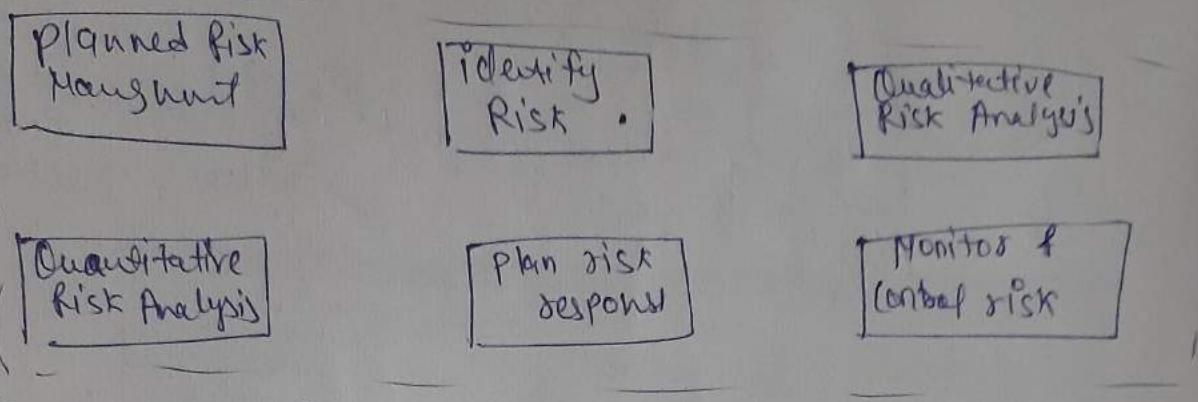
#. Risk Management:- Risk is an uncertain event that may have +ve or -ve impact on project

→ Risk Mangemt is a process of identifying & migrating the risk.

* Why it is important?

Risk affect all aspects of your project i.e budget, schedule, scope.

Risk in Project Management :-



(1) Planned Risk Manage:-
- analysis of decision making to implement R.M.
- Stack holders will be involved in planning risk management
→ Schedule Risk
→ Cost
→ Quality "
→ Scope "
→ Resource "
→ Customer Satisfaction Risk

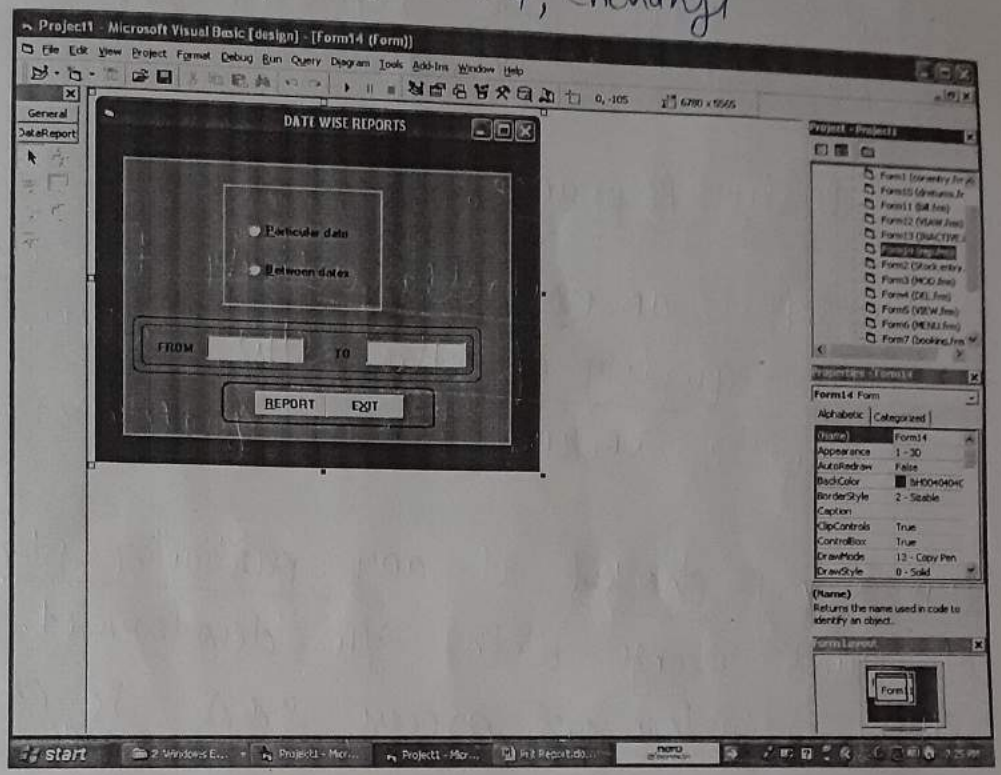
(2) Identify Risk:-
- which risk has more probability of effecting the project?
- Info. gathering
- Checklist Analysis

(3) Qualitative Risk Analysis:-
- assess impact & likelihood of identified risk
- Risk ungency assessment

(4) Quantitative Risk Analysis:-
- data gathering -
= direct, diagramatic
- probability distribution
- Modeling techniques

(5) Plan Risk Response - eliminates threats before they happen

- -ve risk or threat
  → avoid, transfer, mitigate, Accept
- +ve risk or opportunity
  exploit, share, enchang



(6) Monitor & control Risk:-

- risk reassessment
- risk audit
- trend analysis

Project Scheduling:- refers to roadmap of all activities to be done with specified order and within time slot allocated to each activity

for scheduling a project, it is necessary to-

1) Break down the project tasks into smaller, manageable form

2.) find out various tasks & co-relate them.
3.) Estimate time frame required for each task.
4.) Divide time into work-time units
5.) Assign adequate no of work units for each tasks
6) Calculate total time required for the project from start to finish.

S/w Quality Assurance:-

1.) S/w Q.A is an umbrella activity for checking the quality of the s/w, whether the s/w we're making is of quality of not

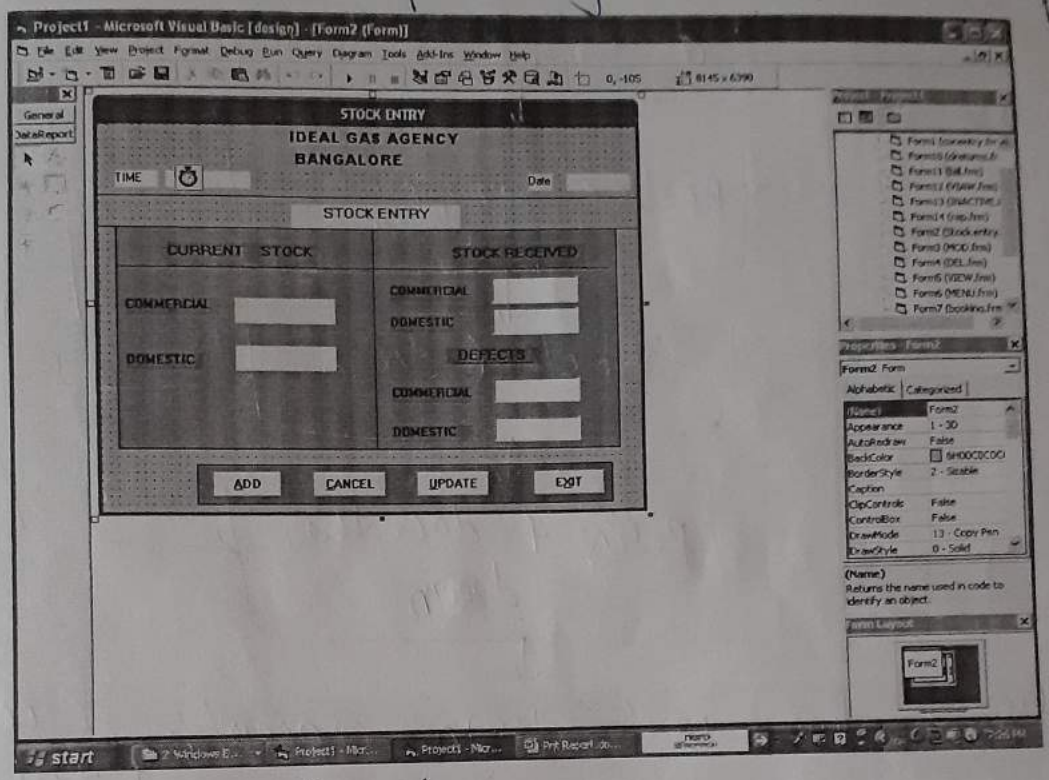
2.) SQA is not checked at any particular phases, It runs alongs with S/w developmet. After every developement process, SQA is checked

3.) generally quality of s/w is verified by 3rd party organization like ISO to ensure s/w must contain Quality product.

4.) A s/w quality product has several quality factors:

- Portability
- Reliability
- functionality
- Error free
- Maintainability
- Usability

* S/w Quality Assurance plan:-
  → outline of quality measures to ensure quality levels.
  → plan is use as baseline to compare actual levels of quality with planned quality level.



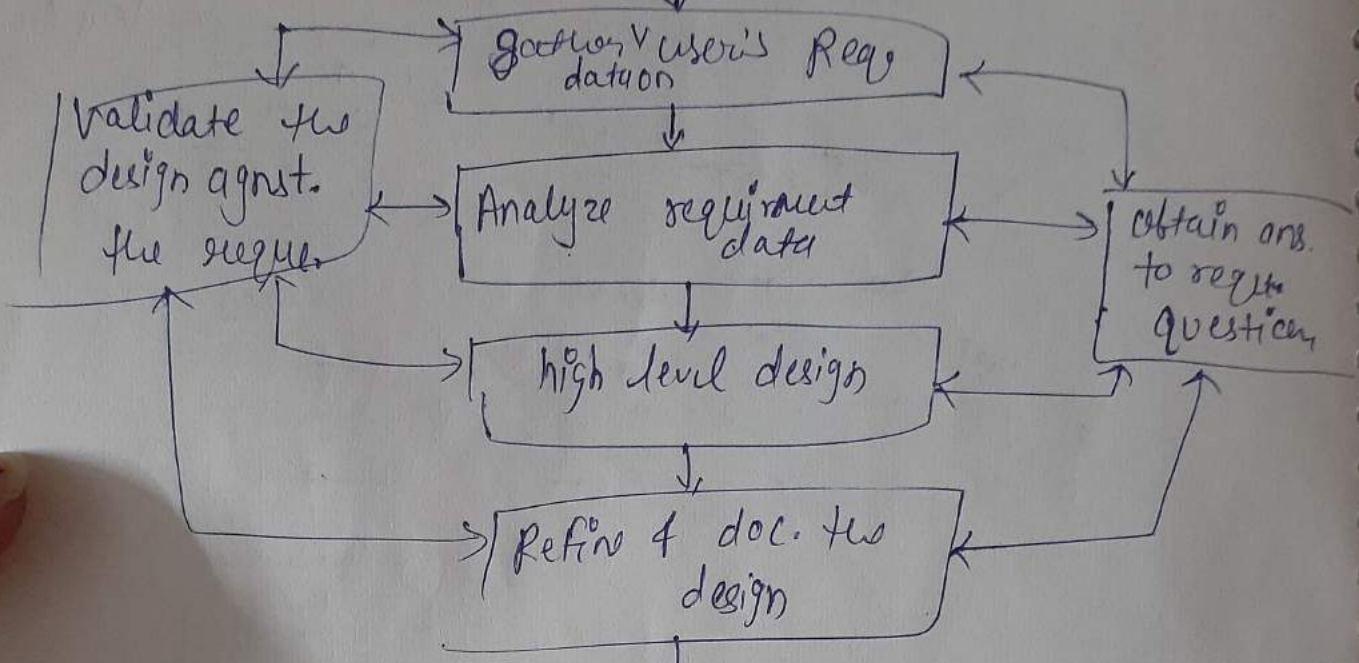
Step1. Document the plan
Step2. Obtain management Acceptance
Step3. Obtain Development Acceptance
Step4. plan for implementation of SQA plan
Step5. Execute the SQA plan

Unit-2

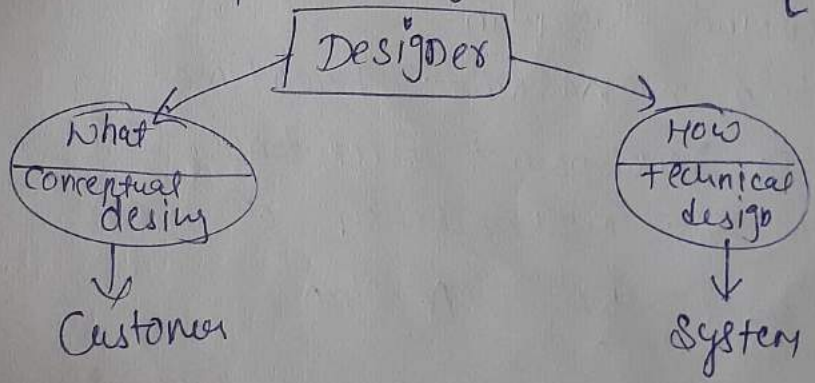
Software Design

⇒ S/w design is more creative than analysis
⇒ problem solving activity

initial requirements



⇒ It's a two part design concept :- { Conceptual, technical }



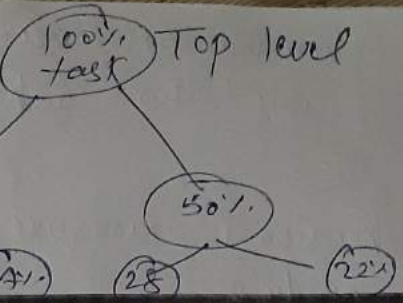
(DFD)

Approaches of S/w design :- of Design Methodology

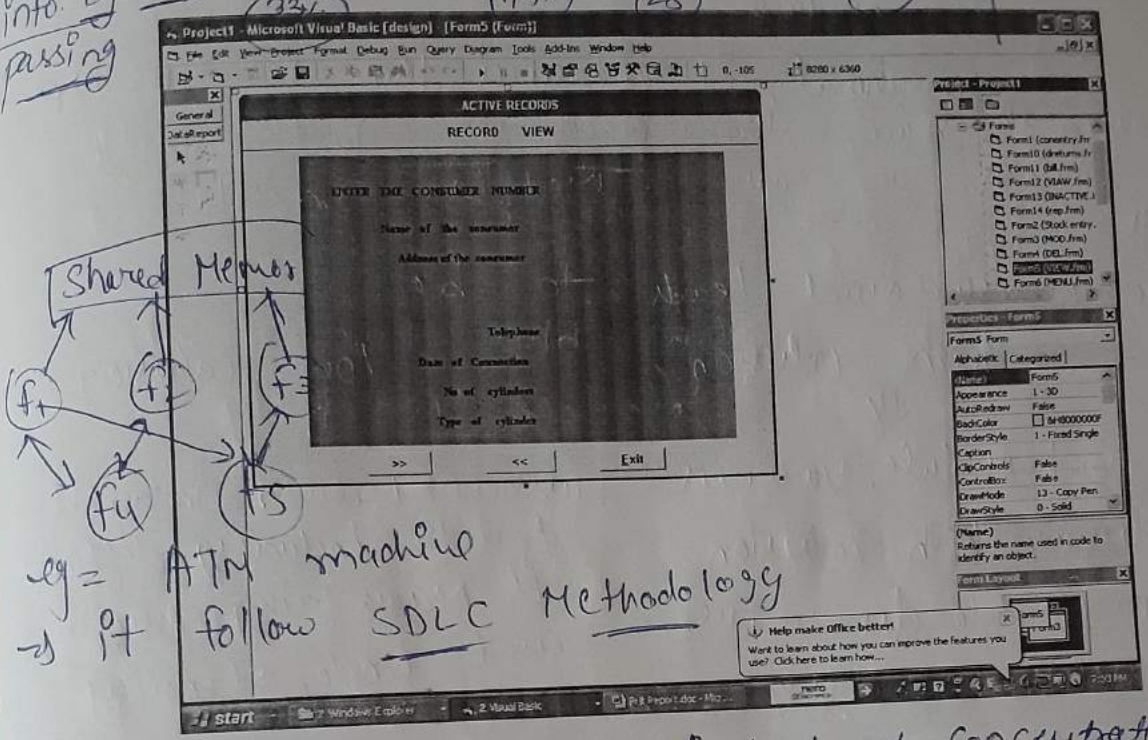
1. function oriented design (FOD)
2. Object oriented design (OOD)

(1.) function Oriented Design :- Top level module is sucessfully decomposed into more sub-module where each module has well-defined task.

⇒ Divide & conquer Method
⇒ tells what system does
⇒ module can share info. by passing info.



⇒ it use
- DFD
- DD
- Structure chart
- Pseudo Code



eg = ATM machine
⇒ it follow SDLC Methodology

(2) Object oriented design :- Instead of concentrating on function we focus on object.
object has data & functions, which're used to create application
⇒ OOD the entire system is viewed as an object
⇒ Each object has it's own state & behaviour
⇒ eg :- TV

Terms :-
- Instance variable :- Data, Attributes, state.
- Instance Method :- operations are specific
- class variables :- data, attribute that are not specific to instance
- Class Method :- operation that're not specific.
- Constructor :- create, initialize object.

Cohension:- Cohesion are necessary in development of any s/w, which is reliable & extendable.

Definition:- Cohesion is measure of functional strength of a module.

→ In general, it measures the relationship strength b/w the pieces of functionality within a given module. In s/w programming.

⇒ The measurement leads to be

high cohesion ⇓ In a good module, various parts having high cohesion is preferable due to it's reliability, reusability robustness & understandability

low cohesion ⇓ associated with undesirable traits including difficult is maintaining reusing and understanding

Advantage of high cohesion:-

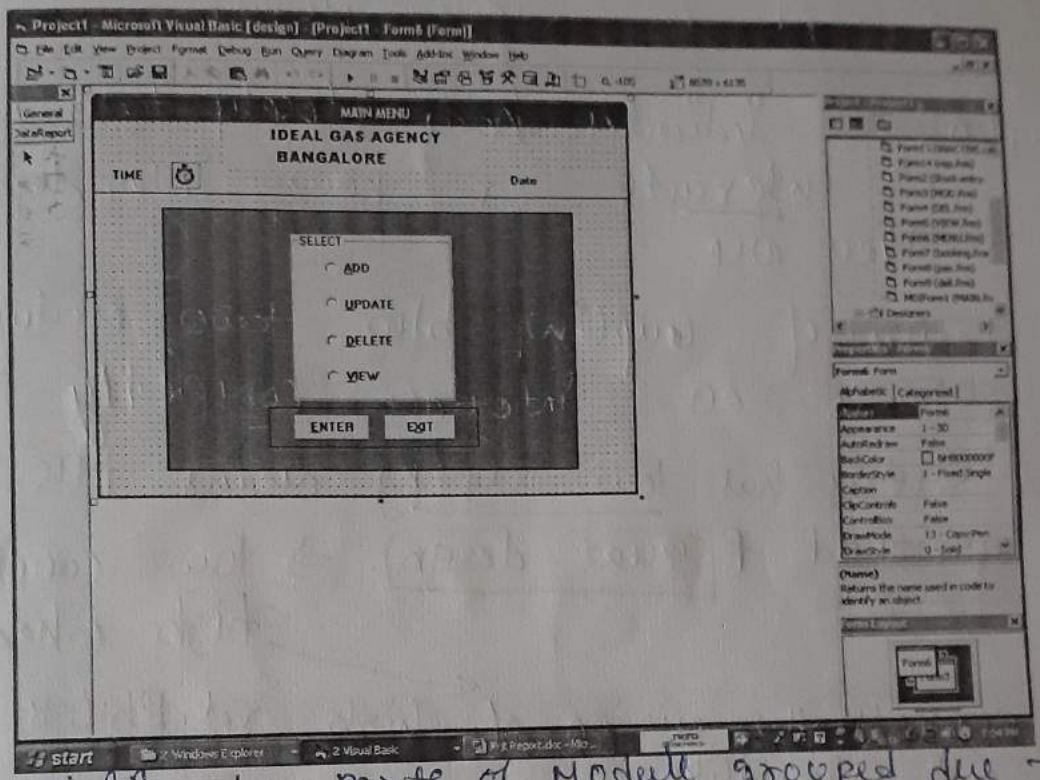
→ Yes Module's reusability
→ System maintainability will be increased.
→ Module complexity also reduces

Classification of cohension

high ↓ functional ↓ Sequential ↓ Communication ↓ procedural ↓ logical ← low

degree of cohesion

(1) functional cohesion:- • Best type of cohesion
• In which part of module are grouped becoz they all contributed to module single well-defined task.



(2) Sequential:- when parts of module grouped dug to the output from one part is the input of the other part

eg:- Sort → search → display

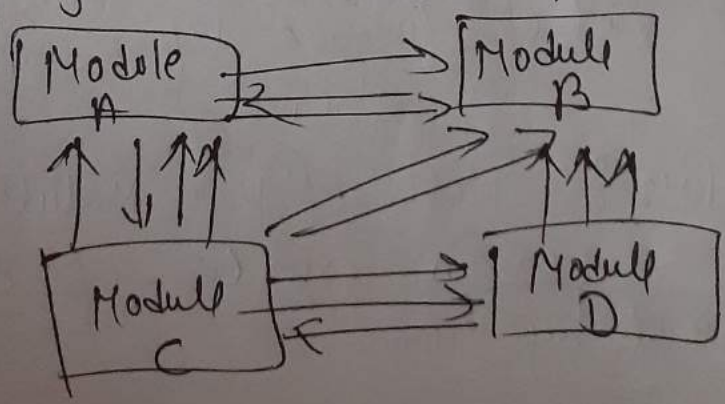
(3) Communication:- The parts of module are grouped becoz they operate on same data
eg:- Set of function defined on an array or stack.

(4) Procedural:- parts of the module are grouped becoz a certain sequence of execution is followed by them.
eg:- Algorithm

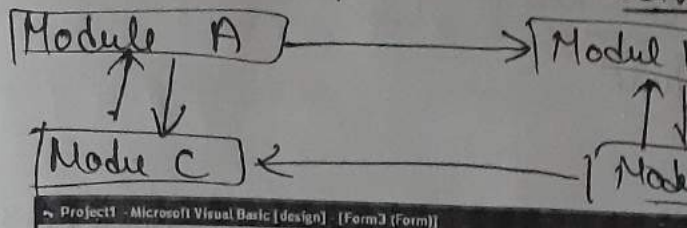
(5) Logical Cohesion:- o worst type of cohesion
All elements of module performs similar operations
eg:- error handling, data i/p etc.
eg:- set of print func to generate o/p report arranged in single module

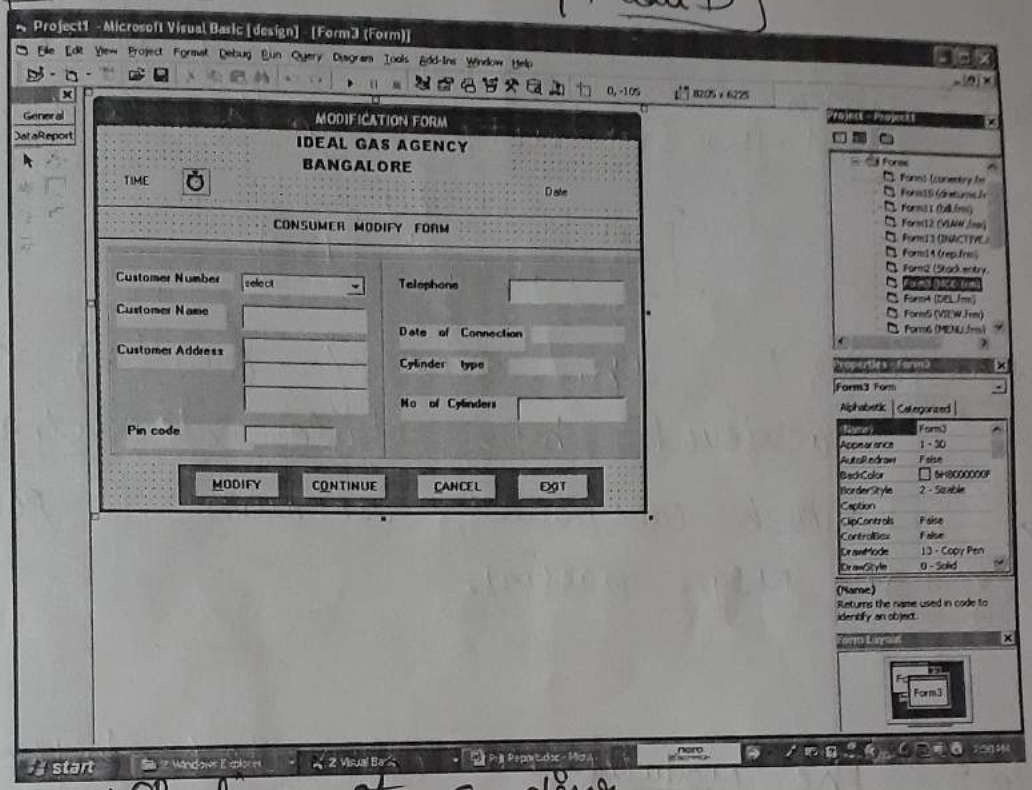
#. Coupling:- It indicates how closely two modules interacts or how inter dependent they are
→ degree of coupling b/w two module depends on interface complexity
→ If system has low coupling means its well structured & great design. ⇒ low coupling high cohension.
→ It supports the mission of high readability & maintainability.
→ Coupling is an indication of strength of interconnection b/w all components in sys.

highly coupled:- System have inter connection, in which program unit depend upon each other



loosely coupled :- Made up of component which is independent or almost independent





* classification of Coupling

high ← low

Content | Common | external | Control | Message passing | data | Stamp

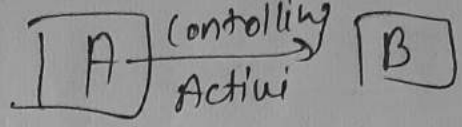
high → low

(1) Content coupling :- occurs when one of Module relays on other Modules Internal working
→ Means a change in 2nd Module, will lead to change in dependent module.

(2) → Common :- aka global coupling
→ It occurs when the same global data are shared by two modules.

(3) External :- It occurs when external imposed data format & common protocal are shared by two Module.

(4) Control :- when one component passes parameter to control the activity of another component.



(5) Message passing :- this type of coupling can be achieved by state decentralization.

→ In which component, communic is performed through msg. passing.

(6) Data :- The modules are connected by data coupling if only data can be passed b/w them

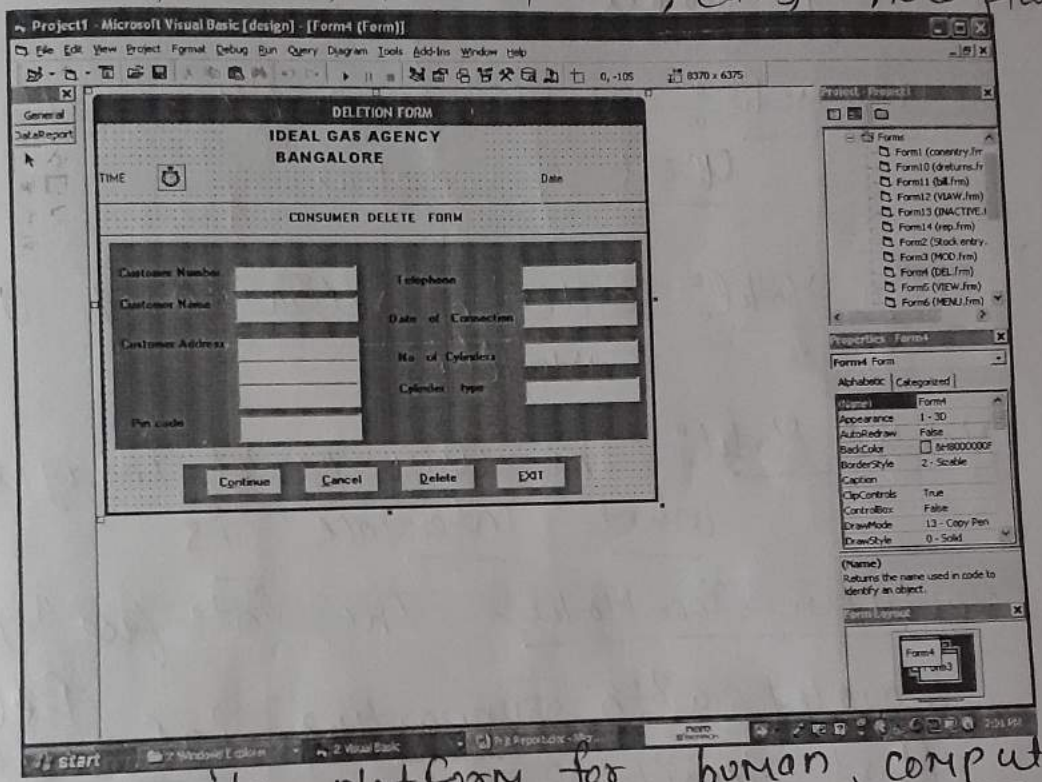
(7) Stamp :- In this type of coupling, data struct. is used to transfer info. from one component to another.

User Interface Design:-

→ U.I is a front end application view to which user interacts in order to use the s/w.

→ User can manipulate & control s/w & H/w by means of user interface.

eg:- Computers, Mobile phone, Cars, Aero plane,



→ It provides platform for human, computer interaction.

→ S/w becomes most popular if its user-interface is

- → Attractive
- → Simple-to use
- → quick response
- → clear to understand

→ UI brodly into 2 categories:

* Command-line Interface
* graph-like Interface

* Golden-Rule of User-Interface = 3 rules

(1.) Place the user in control
(2.) Reduce the user's memory load
(3.) Make the interface consistent

* UI design Models:-

(1.) User Model:- establish a profile of all end users of System

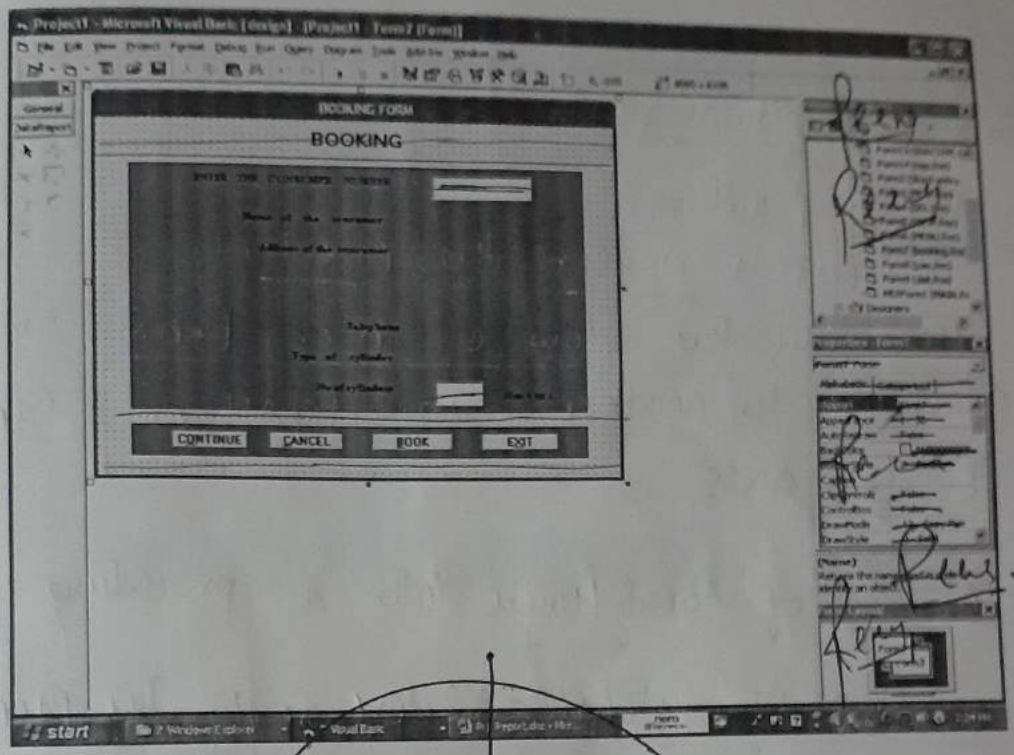
(2.) Design Model:- a design relization of User model.

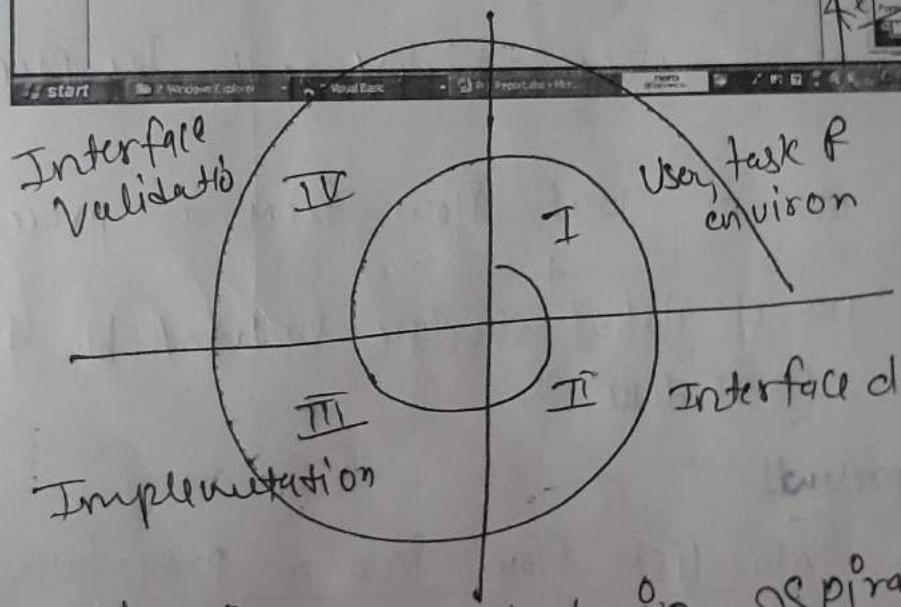
(3.) Mental Model:- The user model images of what interface is

(4.) Implementation Model:- The interface "look & feel" coupled with supporting information that describe interface semantics & syntax.

→ It consists of 4 frame activity:-

1. User task & Environment analysis
2. Interface design
3. Interface construction
4. Interface validation





UI design represented in Spiral Model

Information flow Metrics:-

By Henry & Kafora:-

- attempts to measure the complexity of codes of measuring the flow of info. from one procedure to another in terms of fan ins & fan-outs

Fan-in:- no. of local flows into a procedure
+
no. of global structure ready by procedure

Fan-out:- no. of local flow from a procedure
+
no. of global structure updated by the procedure

Flow Represent:-

- the info. list flow into a procedure via the argument list
- flows from the procedure due to return values of fun^c calls

$$C_p = (f_{an\text{-}in} \times f_{an\text{-}out})^2$$

Complexity of procedure