Date: 02/5/2017

1(c)-Day

Emrana Kabir Hashi

Ref. Book:

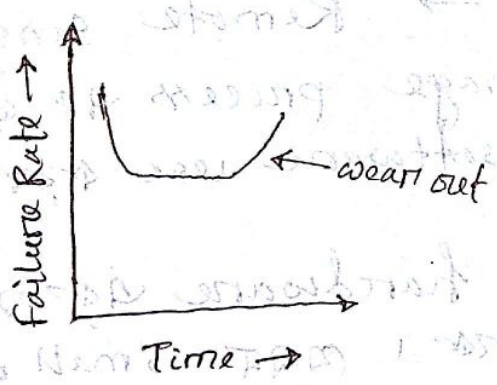
Software Engineering

— Roger Pressman

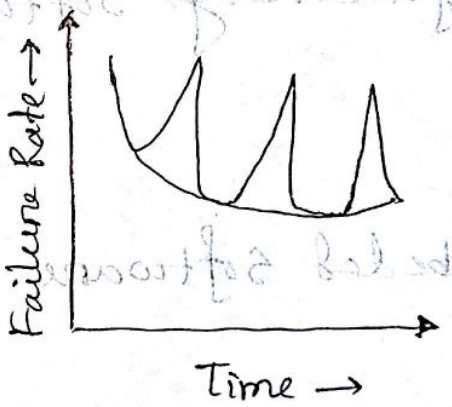
Software → Algorithm
Data structure
Database

Characteristics:

1. s/w is developed or engineered, it is not manufactured in the classic sense.
2. s/w doesn't "wear out" (ক্ষয়প্রাপ্ত হওয়া / অবক্ষয়প্রাপ্ত হওয়া)



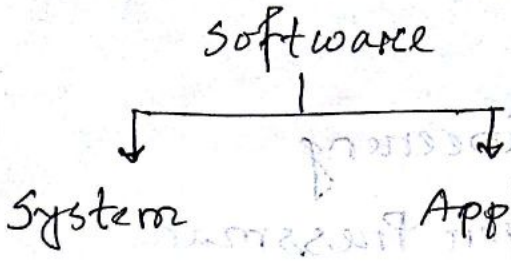
h/w



s/w

3. Although the industry is moving towards component-based assembly, most software continues to be custom built.

1(E)-Day Date: 6/5/2017



System Software → Operating System, Compiler

Application Software → Dependent on the specific application

Business Software → বিভিন্ন company use করে। (যেমন: স্যালারি বা বিদ্যুৎ বিল মেপার জন্য)

Engineering Software → Remote sensing image process এর জন্য যে software use করা হয়।

Imbeded Software → hardware এর সাথে থাকে। (যেমন- small watch)

AI software → Robot

Web based Software → Facebook

Freeware → যে Software গুলো Free তে ব্যবহার করা যায়।

Word Processing Software → MS word

Entertainment Software → Game

2(B) - Day Date: 8/5/2017

Chapter - 2

The Process

→ What process will we apply ?

Layered Technology

→ A Quality Focus
→ Process → Key process area
→ Methods
→ Tools

A Generic View of Software Engineering:

3 phases —

→ 1. Definition phase (? যাবে Question কার আছে Ans
2. Development phase (How →
3. Support
    1. Correction
    2. Addaptation → সব প্রয়োজন মিটে
    3. Enhareancement ~~Interruement~~ কিনা
    4. Prevention → ১, 2, 3 কে support যাবে

Umbrella Activities:

1. Software project tracking and control
2. Formal technical review
3. Software quality assurance
4. Software configuration mangement
5. Document preperation and production
6. Reusability mangement
7. Measurement
8. Risk management

* Developer doesn't consult with end user.
* In what cases customer and end user is same?

2(R) -Day Date: 9/5/

KPA → Key Process Area

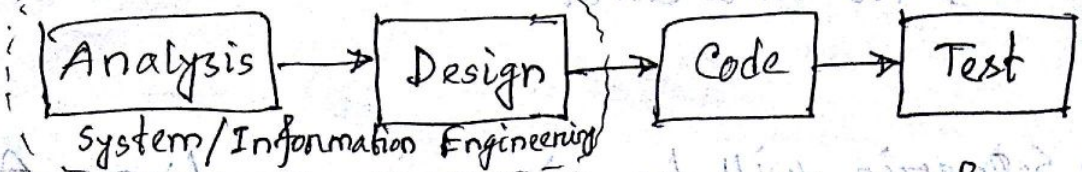
1. Goals
2. Commitments → Requirements
3. Abilities
4. Activity
5. Methods of ~~maint~~ monitoring implementation
6. Methods for verifying implementation

Software Process Model:

→ Linear sequential model
→ Prototyping Model

(waterfall model)

* Linear ~~Sequation~~ ~~Model~~ Sequential Model:



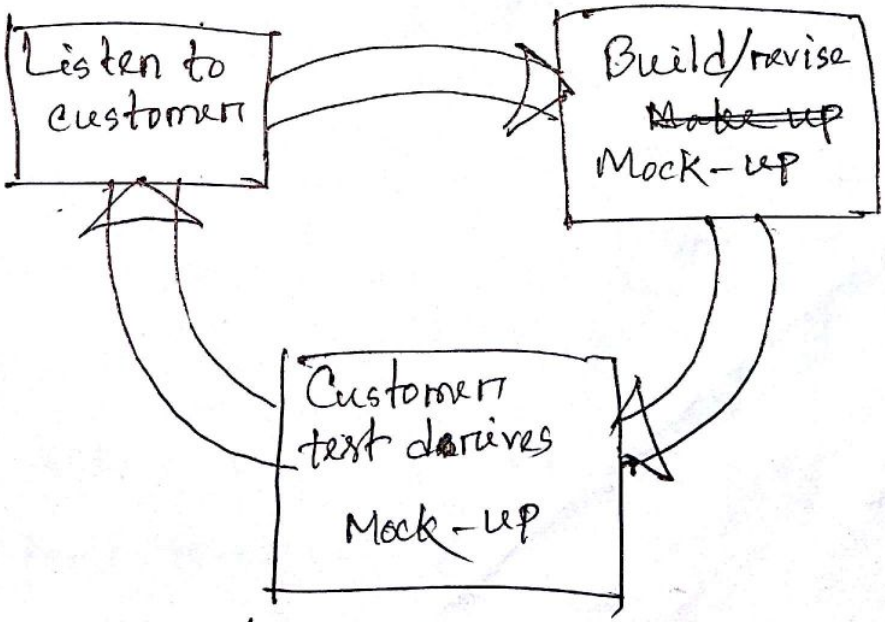
No backtracking way → that's why "waterfall model"

Problem

→ ~~description~~ description

→ where applicable? (When employee and time are not enough)

* Prototyping Model:

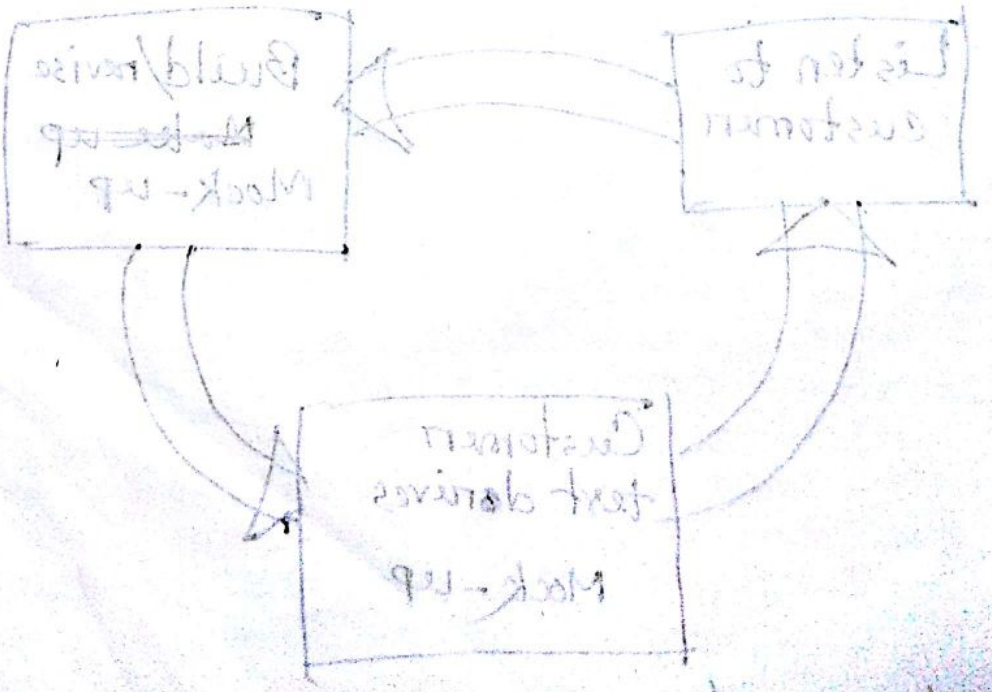


→ When customer is not clear, this ~~pro~~ model is applicable

Problem

→ No time limit

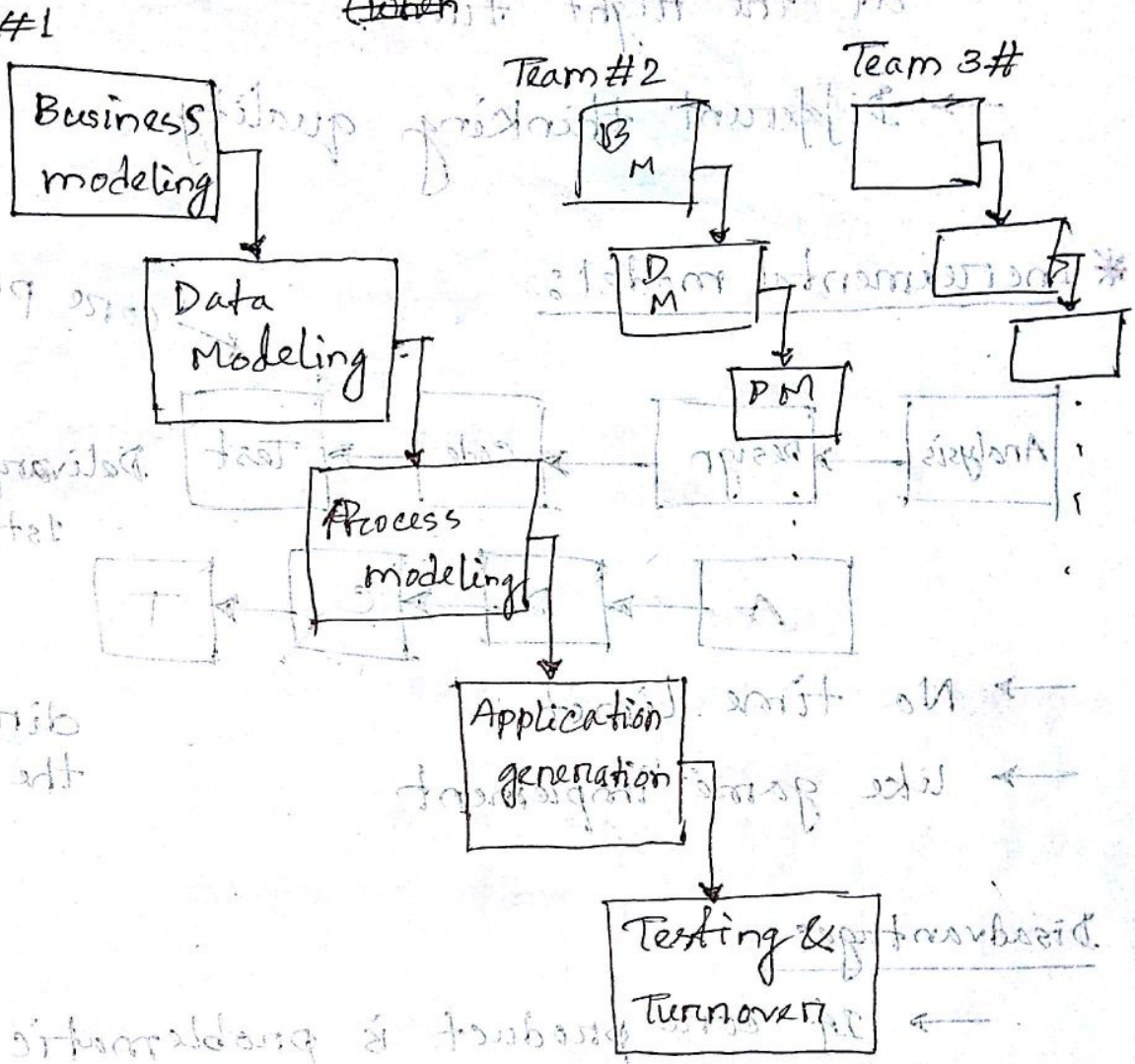
→ Scenario will be given in question. A suitable model has to be picked up according to the requirement of the scienerio.



2(E) - Day 14/8/2017

RAD Model:

RAD → Rapid Application Development



→ when requirement is fully clear and time is limited.

→ When the members are skilled

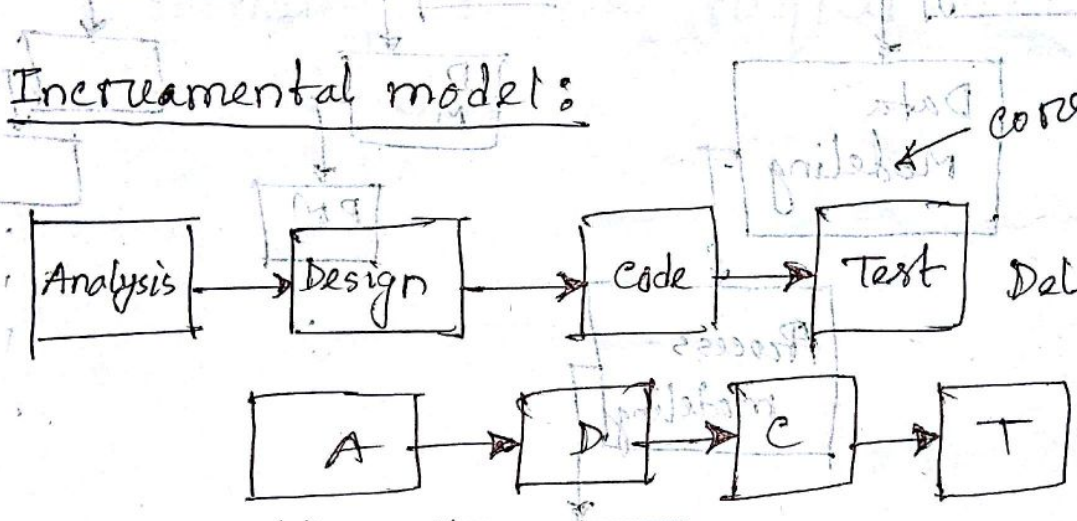
Disadvantage:

→ Huge manpower needed

→ If one team can not deliver their work at the right time.

→ Different thinking quality

* Increamental model:



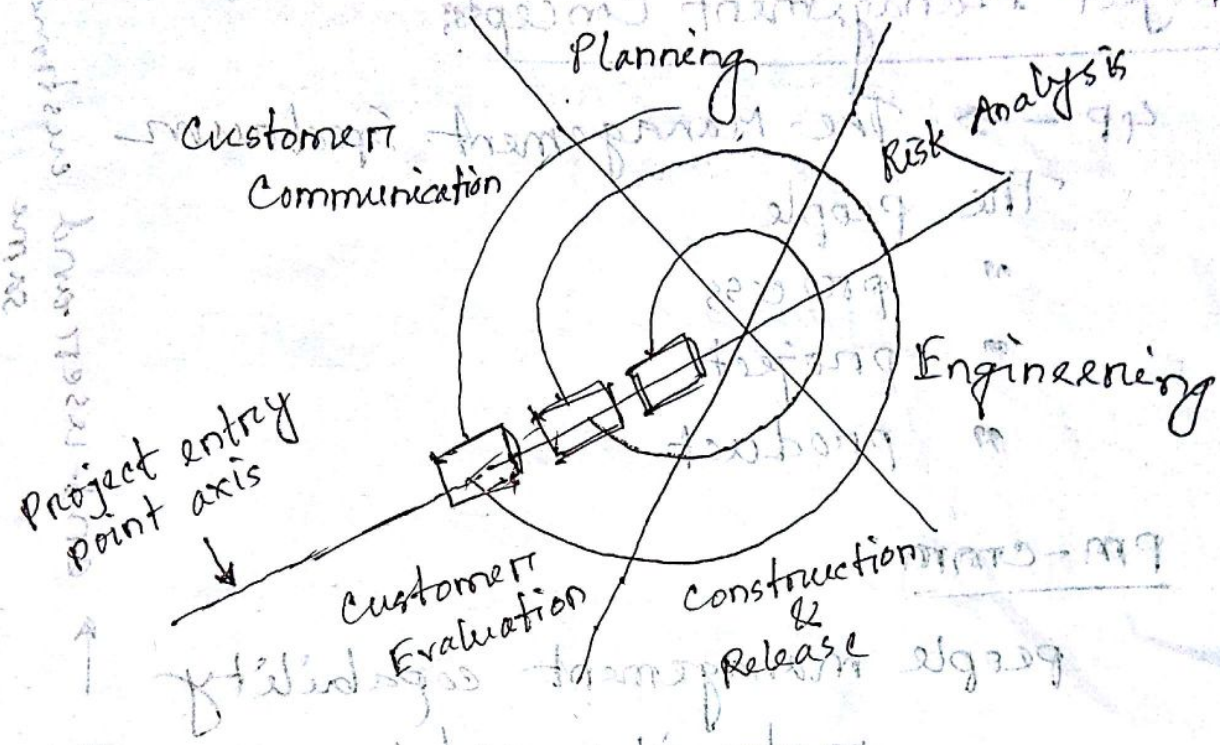
→ No time limit

→ like game implement

Disadvantage:

→ If core product is problematic, then the technical risk cannot be removed.

* Spiral Model:



→ Combination of all models

→ Here 'Risk Analysis' is a unique step

→ Used in a very complicated big project in the corporate firm.

→ Risk analysis must be done well.

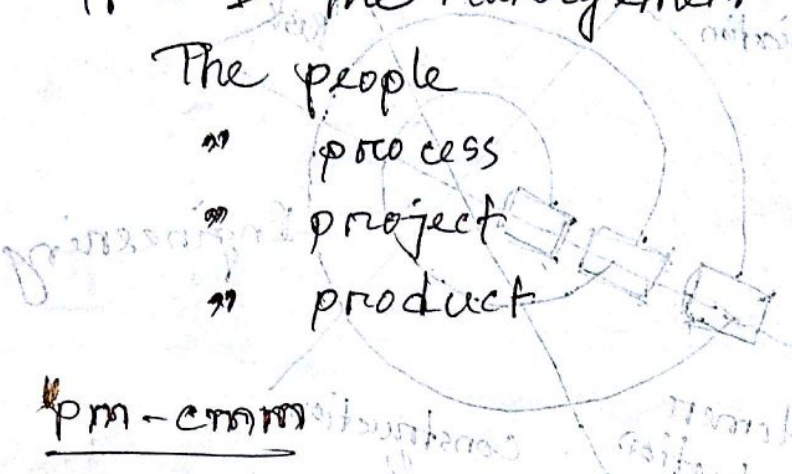
Example & Figure Must

3(B)-Day Date: 16/5/2017

Project Management Concepts:

4P → The Management Spectrum

- The people
- " process
- " project
- " product



pm-cmm

people management capability maturity model

- → s/w people recruiting
- → selection
- → performance management
- → training

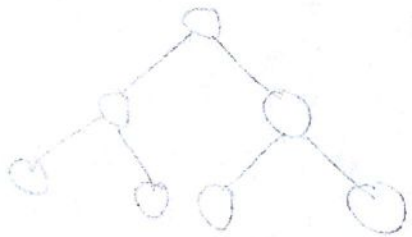
Banking / Railway → end-user and customer same

The players

- → Senior Managers
- → project " / Technical Managers
- → Practitioners
- → Customers / client
- → End-users

Team leader

→ Motivation
→ Organization
→ Idea or Innovation



3(c) - Day

Date : 17/5/2017

Project Manager

- → Problem solving
- → Managerial identity
- → Achievement
- → Influence & team building

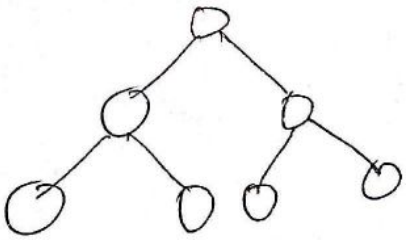
Software Team

Organization

(1) Democratic Decentralized

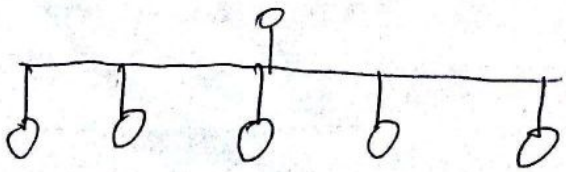
→ when project shald be finished in short time

(2) Controlled Decentralized



→ When team members are more

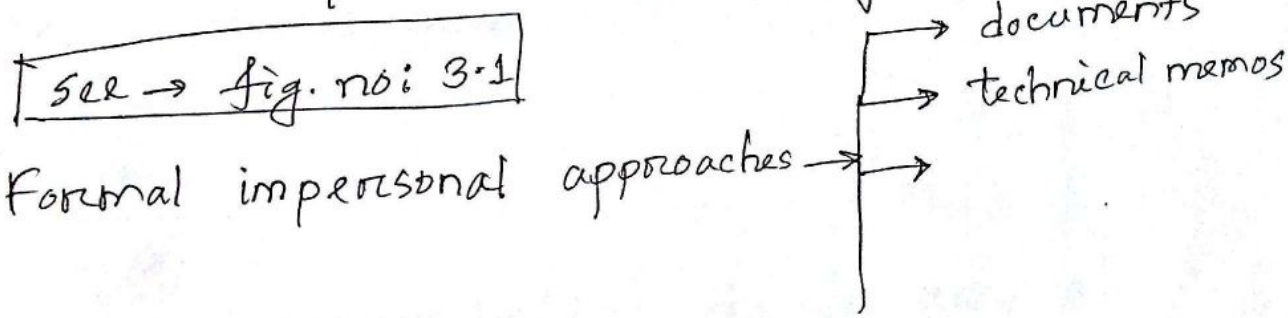
(3) Controlled Centralized



Co-ordination & Communiction Issues:

1. Formal impersonal approaches
2. Formal interpersonal procedures
3. Informal " "
4. Electronic Communication
5. Interpersonal networking

see → fig. no: 3.1



3(E)-Day

Date: 21/5/2017

Product

→ Context
→ Information objectives
→ Function & performance

Process

Various models →

Project

The reasons behind the failure of a project →

→ Software people don't understand the customer's need.
→ The product scope is poorly defined.
→ Changes are managed poorly.
→ The chosen technology changes.
→ Business needs change
→ Deadlines are unrealistic
→ Users are resistant.
→ Sponsorship is lost.

→ The project team lacks people with appropriate skills.

→ Managers avoids ~~best process~~ learning. best practice & lesson learning

The WHHH principle: → WH-questions

Q. Describe any principle

1. Why is the system being developed?
   → for business purpose.
2. What will be done by when?
   → software scope, deadline, schedule making
3. Who is responsible for a function?
   → People related

4. Where are they organizationally located?

→ Team, their location, client, users, customers.

5. How will be the job done technically & managerially?

→ Methodology of the job

6. How much of each resource is needed?

→ skilled person, environment, computers

4(B)-Day

Date: 3/7/2018

Chapter-4

S/W Process and Project Matrices

S/W Measurement ↕ Quantitative Measurement

- Direct
- Indirect

Direct → LOC, execution speed, memory size, defects

Indirect → functionality, quality, complexity, efficiency, reliability, maintainability ...

Error → Finally product release করার আগে team যেটা পায়

Defect → Product release করার পর customer যেসব error পায়

Fig 4.4

Size-Oriented Matrices → for direct measurement

KLOC → p-m → person month → Page of documents

200,000 $

cost

Project	LOC	Effort	$(000)	Pg doc	Error	Defects	People
alpha	1000	24	200	300	150	29	3

4(E)-Day

Date: 8/7/2017

Function oriented Matrices

4.5

Measurement Parameter	Count		Weighting factor Simple	Average	Complex	
No of input	100	×	(3)	4	6	= 300
" " output	50	×	4	(5)	7	= 250
" " inquiries	10	×	(3)	4	6	= 30
" " files	40	×	7	(10)	15	= 400
" " external interfaces	10	×	5	7	(10)	= 100
Count total →						1080

output files (Database files)

Function point,

$$F.P = \text{Count total} \times \left(0.65 + 0.01 \times \sum(F_i)\right)$$

where

f_i $(i = 1 \text{ to } 14)$ complexity adjustment value. (0 to 5)

5(c)-Day Date:

COCOMO (constructive cost model)

Cost estimation
→ At early stage
→ Design stage
→ Project running stage

V.V.I → Exam ची शक्यता !

E → Effort in person-month/year
t → project duration in " "
B → special skills factor
p → productivity parameter

$$E = \left[LOC \times B^{0.333} / p\right]^3 \times \left(1/t^4\right)$$

→ Real time embeded software
p = 2000

→ Telecommunication & system software
p = 10000

→ Business system app
p = 8000

→ LOC (5000-15000) → B = 0.16

→ LOC (70000 or above) → B = 0.39

→ LOC (15000 – 70000) → B = 0.16 এবং 0.39 এর মাঝামাঝি ধরতে হবে।

Person-month → round figure দিতে হবে। এর উপরে হলে ceil আর নিচে হলে floor

ক (E)-Day

Date: 15/7/2017

Chapter-6

6.1 RISK ANALYSIS AND MANAGEMENT

Risk Strategy

→ Reactive → যখন risk হবে, তখন দেখা যাবে

→ Proactive → আগে থেকেই risk নিয়ে চিন্তাভাবনা করা

Difference → Exam এ আসতে পারে

6.2 Two characteristics of software risk

→ Uncertainity

→ Loss

Risk catagories

→ Project

→ Technical

→ Business

→ ~~Load~~ known

Project risk

- → Potential budgetary
- → Schedule
- → Personnel
- → resource
- → customer
- → requirement problems

Technical risk

- → Potential design
- → Implementation
- → Interface
- → verification
- → Maintanance problems

Date: 18/7/2017

6(C)-Day

Business Risk →
- → Market, → মার্কেটে চলবে কি না
- → strategy
- → Management
- → Budget

Known Risk → তিন ধরনের (Project, Technical, Business) Risk এর যদি uncoverable হয়।
- → predictable (poor communication. —
- → Unpredictable

Risk Identification

Predictable risk can only be identified

- → Generic risk
- → Product specific risk

Ho

→ Product size
→ Business impact
→ Customen characterics
→ Process definition
→ Development environment
→ ~~Technology~~ Technology to be built
→ Staff size and experience

RMMM

Risk mitigation monitoring & management

→ Risk avoidance
→ Risk monitoring
→ Risk mangement

6(E)-Day 1 (8) F Date: 29/7/2017

Chapter-7

Project Scheduling

Project cannot be submitted in time. Because-

1. Unrealistic deadline
2. change of customer's requirement
3. shortage of resources
4. Technical difficulties
5. Miscommunication

- ~~Degree of~~

7(B) - Day Date: 31/7/2017

7.3.1 Degree of Rigor

→ ~~Casual~~ Casual
→ structure
→ strict
→ Quick reaction

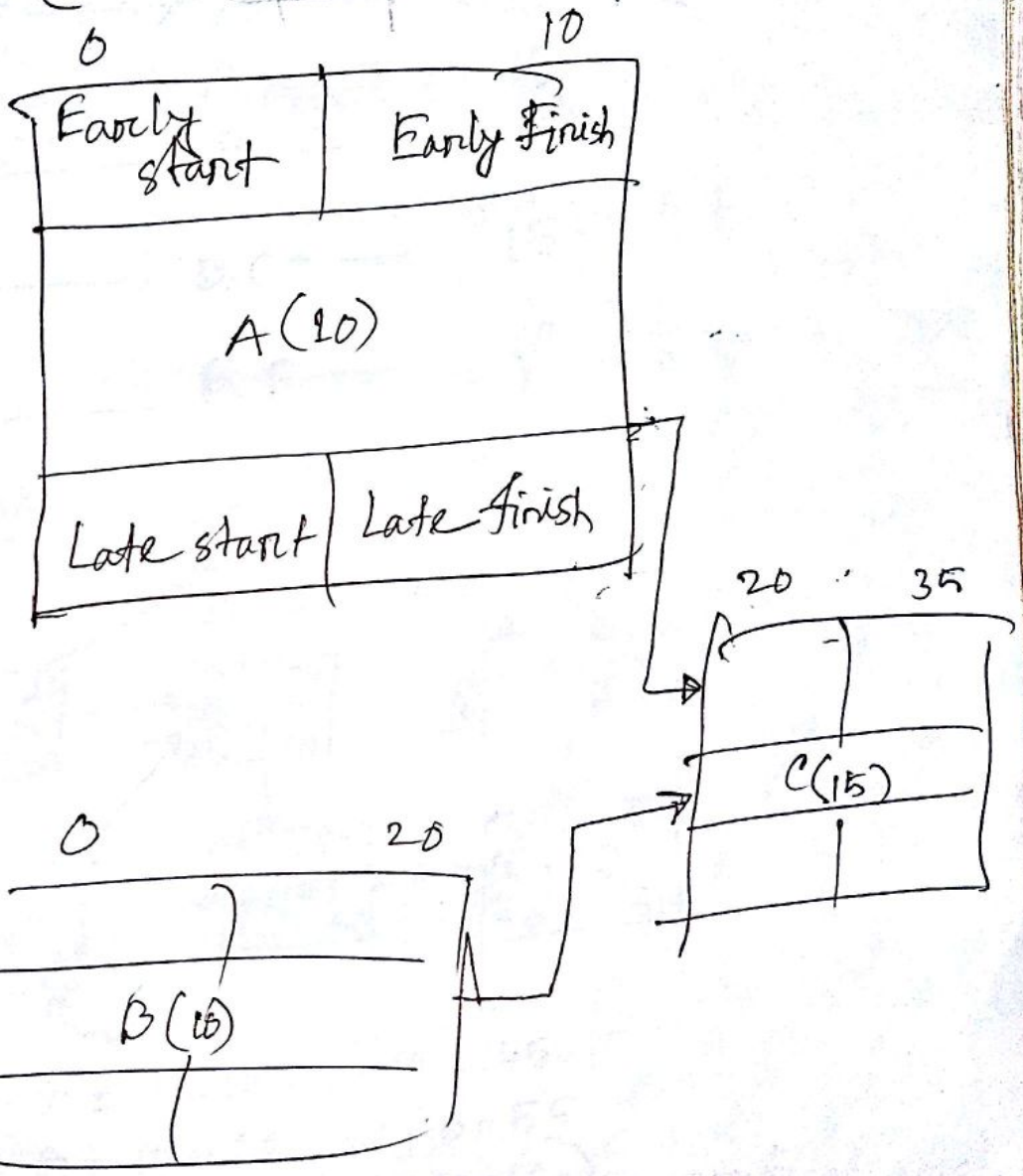
7.6 Task network figure

slide থেকে
↓
S/W scheduling

Network diagram
→ CPM → Critical Path Method (deterministic)
→ PERT → Program Evaluation and Review Technique (Probabilistic)

Bar diagram
→ Gantt Chart

Task	Predecessors (dependent on)	Time
A	—	10 d
B	—	20 d
C	— A, B	15 d
D	— C	



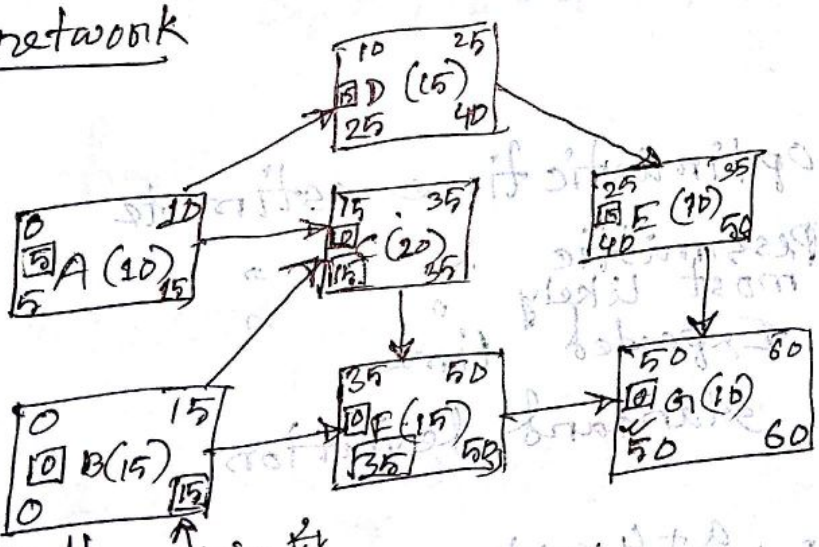
7(c)-Day

Date: 01/8/2017

CPM

Task	Predecessor	Time (d)
A	—	10
B	—	15
C	A, B	20
D	A	15
E	D	10
F	B, C	15
G	E, F	10

Task network



Critical Path

A-D-E-G = 10 + 15 + 10 + 10 = 45

A-C-F-G = 10 + 20 + 15 + 10 = 55

B-C-F-G = 15 + 20 + 15 + 10 = (60) ← critical path

B-F-G = 15 + 15 + 10 = 40

Disadvantage → Task বেশি হলে network complex হয়।

Slack time

Slack time = Late start − early start

Critical path এর task গুলোর কোন slack time থাকবে না।

* Critical path এর standard deviation বের করতে হবে।

PERT

a = optimistic time estimate
b = Pessimistic " "
m = most likely " "
t_e = Expected time
s_i = standard deviation

$$t_e = \frac{a+4m+b}{6}$$

$$s_i = \frac{b_i - a_i}{6}$$

$$S_{cp} = \sqrt{s_1^2 + s_2^2 + \dots + s_n^2}$$

At $\begin{cases} m = 10 \\ a = 9 \\ b = 12 \end{cases}$ $t_e = \frac{a+4m+b}{6} = \frac{9+40+12}{6}$

$= \frac{61}{6} = 10.16$

$S_A = \frac{12-9}{6} = 0.5$

$(10.2 \pm 0.5) = 9.7 \sim 10.7$

H.W

$S_{cp} = ?$

B - C - F - G = 60

$S_B \quad S_C \quad S_F \quad S_G$

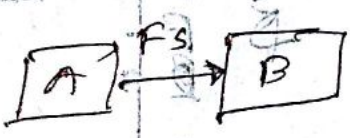
$S_{cp} = \sqrt{S_B^2 + S_C^2 + S_F^2 + S_G^2}$

Bar chart

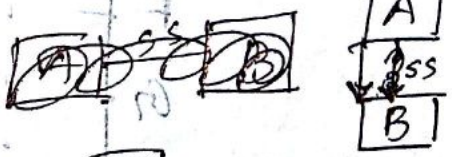
Gantt Chart

Dependency types

1) Finish to start

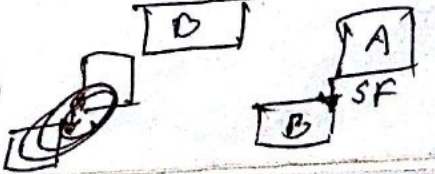


ii) start to start

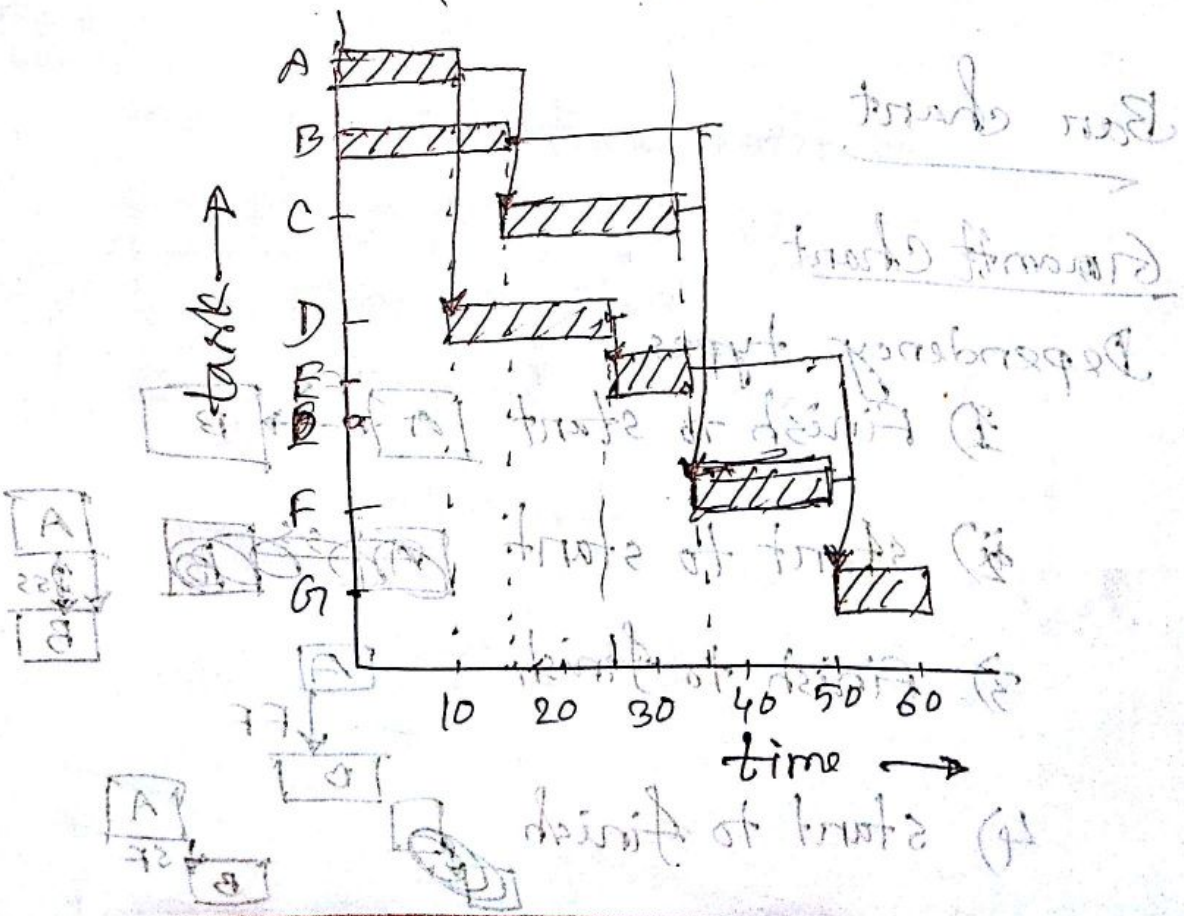


3) Finish to finish

4) start to finish



Task	Predecessor	time
A	—	10
B	—	15
C	A, B	20
D	A	15
E	D	10
F	B, C	15
G	E, F	10



7(E)-Day Date: 5/8/2017

Chapter-8

Software Quality Assurance

→ Quality of Design
→ Quality of ~~Performance~~ Conformance

User satisfaction

Quality Control

→ Inspection
→ ~~Refuse~~ Reviews
→ Test

Quality Assurance

→ Documentation

Cost of Quality

CT-2

Chapter-4

There are four reasons for measuring software processes, product and resources-

→ To characterize
→ To evaluate
→ To predict
→ To improve

Measures:

When a single data point has been collected, a measure has been established.

Measurement:

Measurement ~~result~~ occurs as the result of the collection of one or more data points.

Metrics

A software metric relates the individual measures in some way.

Indicator:

An indicator ~~provides~~ is a metric or combination of metrics that provide insight into the software process, a software project or the product itself.

Importance of software process and project metrics:

With measurement,

- → trends can be spotted.
- → better estimates can be made.
- → true improvement can be accomplished over time.

Software measurements in the physical world can be categorized in two ways -

- → Direct measures
- → Indirect measures

* Direct measures includes
  - → cost and effort applied
  - → lines of code (LOC) produced
  - → execution speed
  - → memory size
  - → defects reported over some set period of time.

* Indirect measures includes
  - → functionality
  - → quality
  - → complexity
  - → efficiency
  - → reliability
  - → maintainability

and many otherr abilities.

Size Oriented Metrics

- Size oriented software metrics are derived by normalizing quality and/or productivity measures by considering the size of the software that has been produced.

Project	LOC	Effort	$(000)	Pp.doc	Errors	Defects	People
alpha	12,100	24	168	365	134	29	3

→ Errors per KLOC
→ Defects per KLOC
→ $ per KLOC
→ Page of documentation per KLOC
→ Errors per person-month
→ LOC per person-month
→ $ per page of document

Function Oriented Metrics:

Function oriented software metrics use a measure of the functionality delivered by the application as a normalization value. Functionality cannot be measured directly. It must be derived indirectly using other direct measures.

Function points are derived using an impirical relationship based on countable measures of software's information domain and assessments of software complexity.

Measurement parameter	Count		Weighting factor simple	average	complex.	
Number of user inputs	100	x	(3)	4	6	= 300
" " " outputs	50	x	4	(5)	7	= 250
" " " inquiries	10	x	(3)	4	6	= 30
" " lines	40	x	7	(10)	15	= 400
" " external interfaces	10	x	5	7	(10)	= 100
Count total						1080

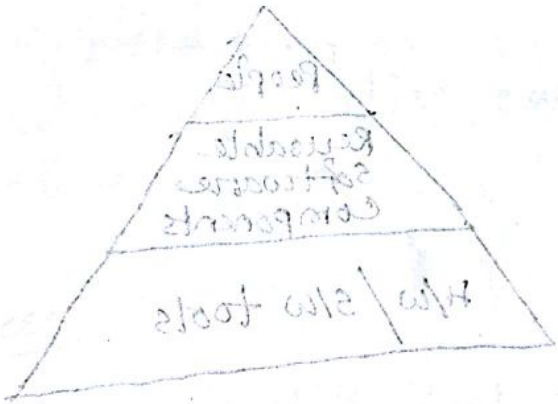
Function Point

Based on 14 questions ↓ on page 91

$FP = \text{count total} \times [0.65 + 0.01 \times \Sigma(F_i)]$

$= 1080 \times [0.65 + 0.01 \times 50]$ $\quad \Sigma(F_i) = 50$ [estimated]

$= 55242$



Chapter-5

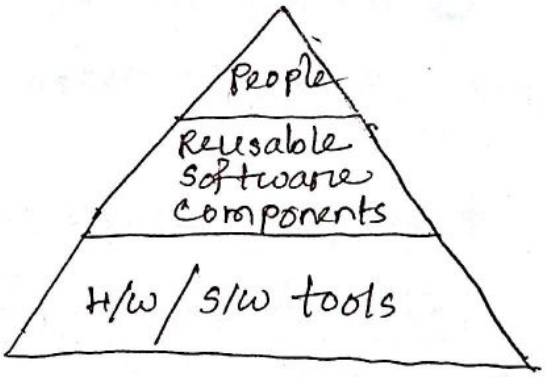
Software Project Planning

Observations on Estimating:

Estimation carries inherent risk and this risk leads to uncertainty

- → Project complexity
- → Project size
- → Degree of structural uncertainty

Resources



- → H/w and s/w tools sits at the foundation of resources pyramid and provides the infrastructure to support the development effort.
- → Reusable software components - software building blocks that can dramatically reduce development costs and accelerate delivery.

→ At the top of the pyramid is the primary resource—people.

Each resource is specified with four characteristics:

→ description of the resource
→ a statement of availability
→ time when the resource will be required
→ duration of time that resource will be applied.

The last two characteristics can be viewed as a time window.

* Human Resources

→ The number of people required for a software project

* Reusable Software Resources

Bennatan suggests four software resource catagories that should be considered as planning proceeds:

* Off-the-shelf components:

Existing software that can be acquired from a third party or that has been developed internally for a past project.

* Full-experience components:

Existing specifications, designs, code, or test data developed for past projects that are similar to the software to be built for the current project.

* Partial-experience components:

Existing specifications, designs, code, or test data developed for past projects that are related to the software to be built for the current project but will require substantial modification.

* New components:
Software component that must be built by the software team specifically for the needs of the current project.

* Environmental Resources:
The environment that supports the software project, often called the software engineering environment (SEE), incorporates hardware and software.

The COCOMO (Constructive Cost Model) Model:
COCOMO II is actually a hierarchy of estimation models that address the following areas.
→ Application composition model
→ used during the early stages of software engineering.
→ Early design stage model
→ Used once requirements have been stabilized and basic software architecture has been established.
→ Post-architecture-stage model
→ used during the construction of the software

* The Software Equation

$$E = [LOC \times B^{0.333} / P]^3 \times (1/t^4)$$

where

E = Effort in person-month/year

t = project duration in months or year

B = special skills factor

P = productivity parameter

For P

→ Real time embeded software

P = 2000

→ Telecommunication & system software

P = 10000

→ Business system application

P = 8000

For B

→ LOC (5000 - 15000), B = 0.16

→ LOC (70000 or above), B = 0.39

→ LOC (15000 - 70000), B = at the middle of 0.16 and 0.39

Chapter-6

RISK ANALYSIS AND MANAGEMENT

Risk strategy

- Reactive
- Proactive

→ Reactive strategy monitors the project for likely risks. This strategy is applied when the risk appears.

→ A considerably more intelligent strategy for risk management is to be proactive. A proactive strategy begins long before technical work is initiated.

Two characteristics of software risk

- Uncertainty
- Loss

→ Uncertainty:
The risk may or may not happen; that is, there are no 100% probable risks.

→ Loss:
If the risk becomes a reality, unwanted consequences or losses will occur.

* Project risks:

Project risks threaten the project plan. If project risks become real, it is likely that project schedule will slip and that costs will increase. Project risks identify-

→ Potential budgetary
→ schedule
→ Personnel
→ Resource
→ customer
→ requirement problems and their impact on a software project.

* Technical risks:

Technical risks threaten the quality and timeliness of the software to be produced. If a technical risk becomes reality, implementation may become difficult or impossible.

Technical risks identify

→ Potential design
→ Implementation
→ Interface
→ Verification
→ Maintenance problems

* Business risks

Business risks threaten the viability of the software to be built

→ Market risks
→ Strategic risks
→ Management risks
→ Budget risks

* Known risks

Known risks are those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information sources:

* Risk Identification

→ Risk identification is a systematic attempt to specify threats to the project plan. There are two distinct types of risks –

- → Generic risks
- → Product specific risks

→ Generic risks are a potential threat to every software project

→ Product specific risks can be identified only by those with a clear understanding of the technology, the people, and the environment that is specific to the project at hand.

* Risk item checklist

- → Product size
- → Business impact
- → Customer characteristics
- → Process definition
- → Development environment
- → Technology to be built
- → Staff size and experience

* The RMMM Plan

RMMM → Risk Mitigation, Monitoring and Management

The RMMM plan documents all work performed as part of risk analysis and is used by the project manager as part of the overall project plan.

9(B) - Day Date: 16/8/2017

Chapter-8

8.14

Prevention Costs

→ Quality planning
→ Formal technical reviews
→ test equipment
→ training

Appraisal →

→ In process & interprocess inspection
→ Equipment calibration and maintenance.
→ testing

Internal

→ rework
→ repair
→ failure mode analysis

External

→ complaint resolution
→ product return & replacement
→ help line support
→ Warranty work

6.3.2 SQA - activities

point গুলো লেখতে হবে।

→ Prepares an SQA plan for a project

→

Chapter - 7

Software Configuration Management

→ Identify change
→ Control change
→ Ensure proper implementation of change
→ Report changes to consult people

9(E)-Day Date: 21/8/2017

Software configuration Management

Process:
Five Tasks →
→ Identification → at design section
→ Version control ← Imp
→ Change Control
→ Configuration Auditing
→ Reporting

* Version Control

Major . Minor . Patch

Major → Incompatible API changes
Minor → Add functionality in a backward compatible manner
Patch → Backward compatible bug fixes

Example should be given

V1 1.0.0
Major Minor patch

V2 1.1.0 [If only minor changes]
V3 1.2.0
V5 2.2.1

→ Pre-released version

→ Metadata (Knowledge about knowledge)

1.0.0 – α (pre-released version)
– 3.1

1.2.0 + .001 (Metadata)

1.0.0 – α + .001 (Metadata of pre-released version)

Chapter-10

System Engineering

→ Software

→ Hardware

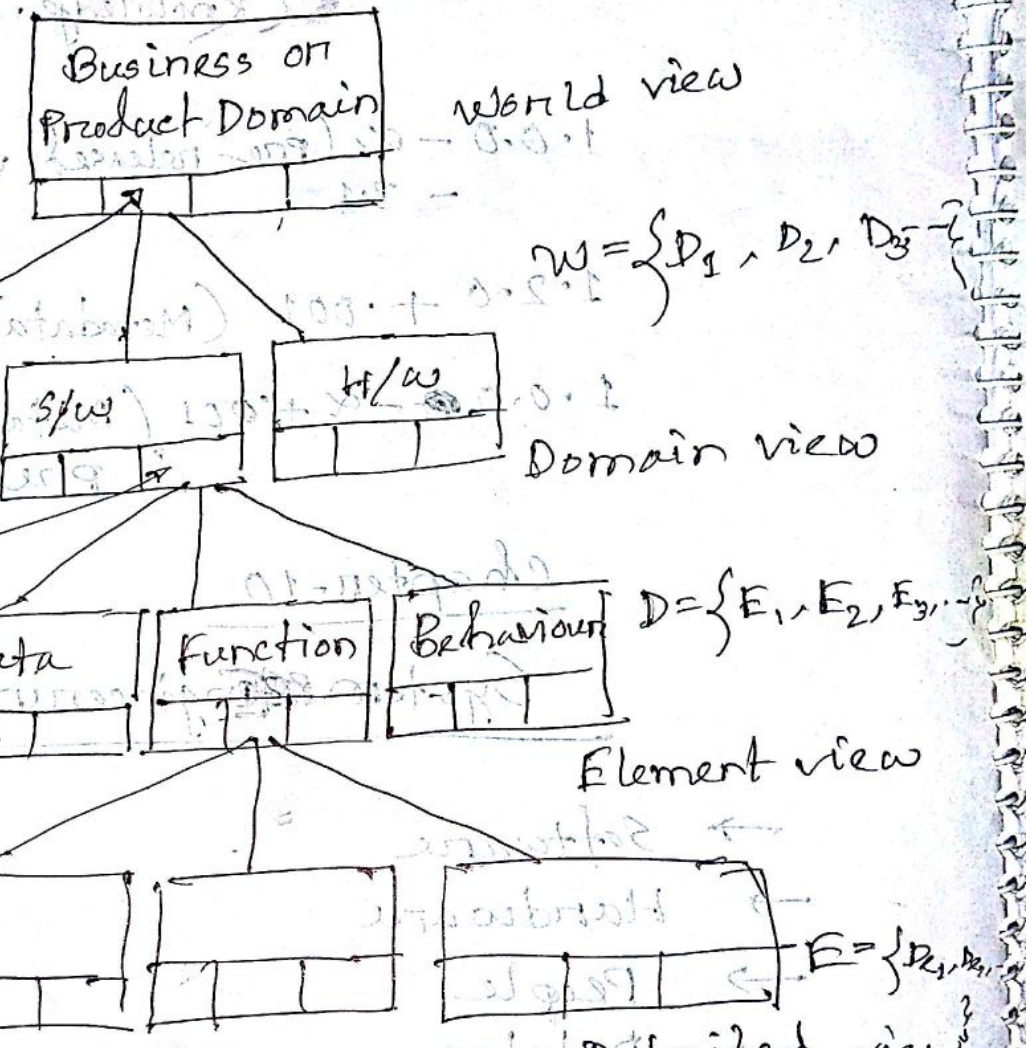
→ People

→ Database

→ Documentation

→ Procedure

System Engineering Element Hierarchy



Book
Fig. 10.3

CT → 7, 8, 9, 10

#CT-3

Chapter 7

Reasons lying behind late software delivery:

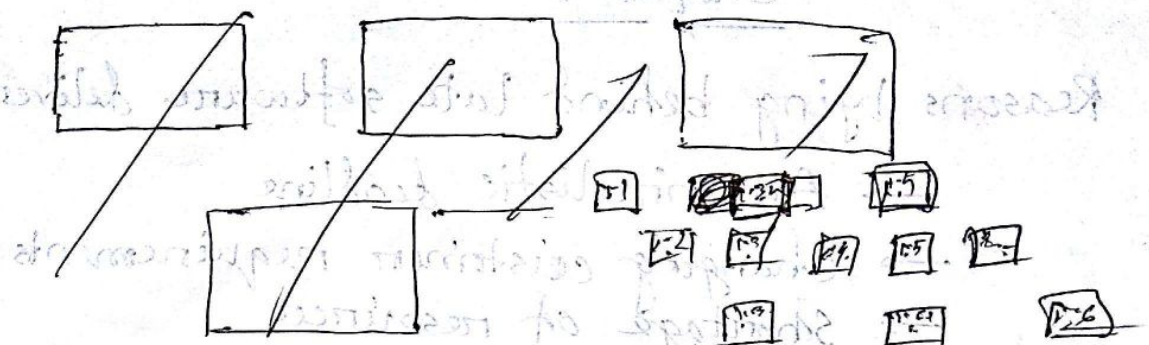
- → An unrealistic deadline
- → Changing customer requirements
- → Shortage of resources
- → Technical difficulties
- → An honest underestimate of the amount of effort that will be required to do the job
- → Human difficulties
- → Failure by project mangement
- → Miscommunication among project stefff

Degree of Rigor

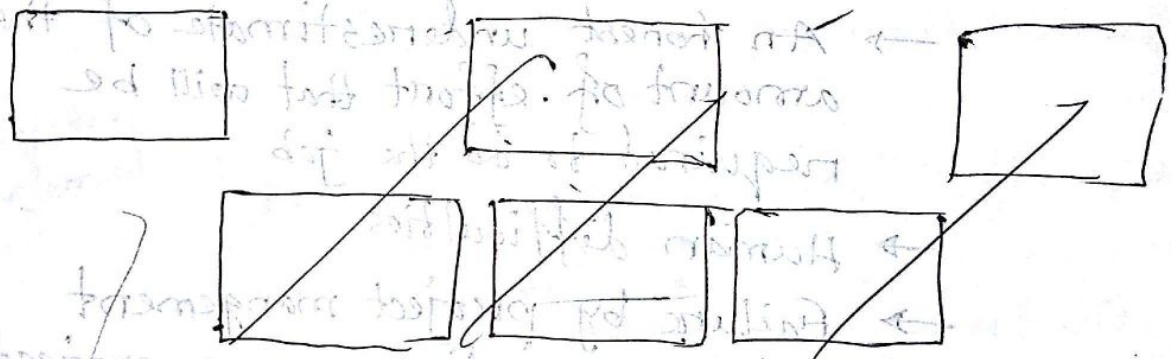
The degree of rigor with which is a function of many project characteristics. Focen different degrees of rigor can be defined.

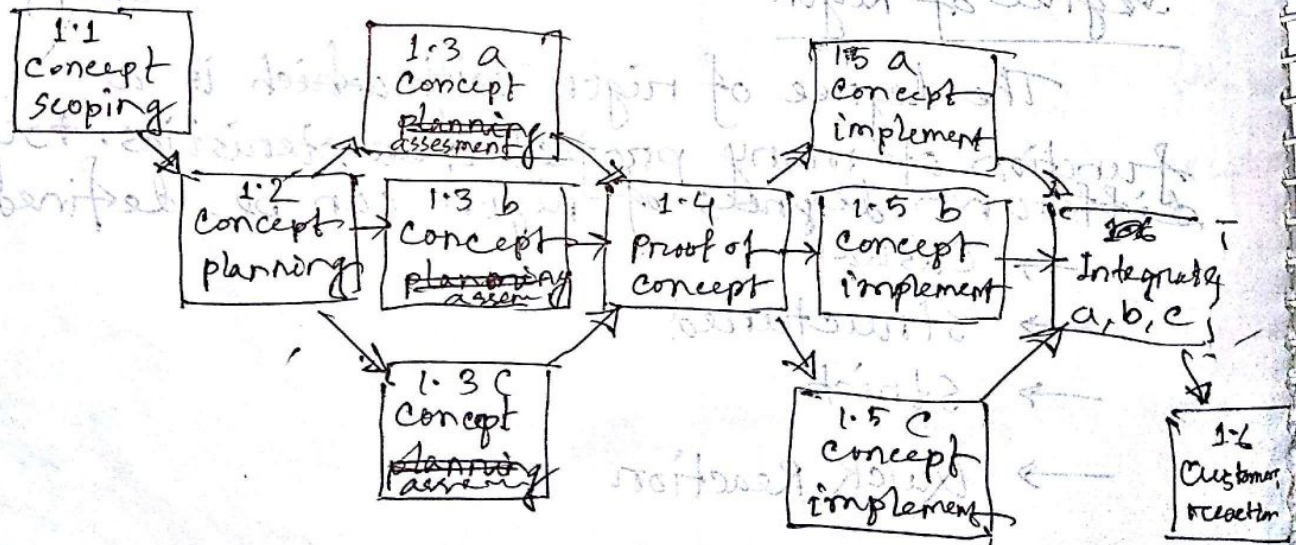
- → Casual
- → structured
- → Strict
- → Quick Reaction

Task Network Diagram:



See page → 181





Network Diagram

Two project scheduling Method —

→ CPM (Critical Path Method)

→ PERT (Program Evaluation and Review Technique)

Bar Diagram

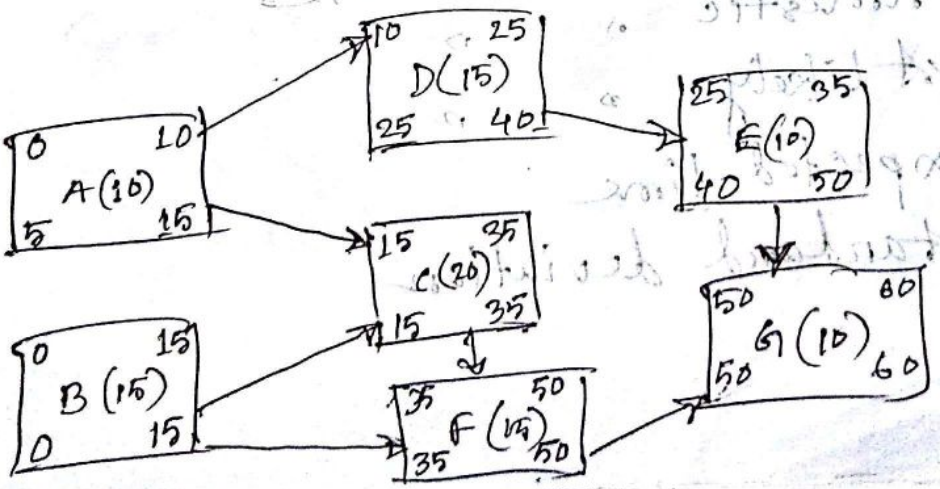
→ Gantt chart

CPM

Table-1

Task	Predecessor	Time (d)
A	—	10
B	—	15
C	A, B	20
D	A	15
E	D	10
F	B, C	15
G	E, F	10

Task Network



Path

A-D-E-G = 10+15+10+10 = 45
A-C-F-G = 10+20+15+10 = 55
B-C-F-G = 15+20+15+10 = 60
B-F-G = 15+15+10 = 40

So the critical path is

B-C-F-G

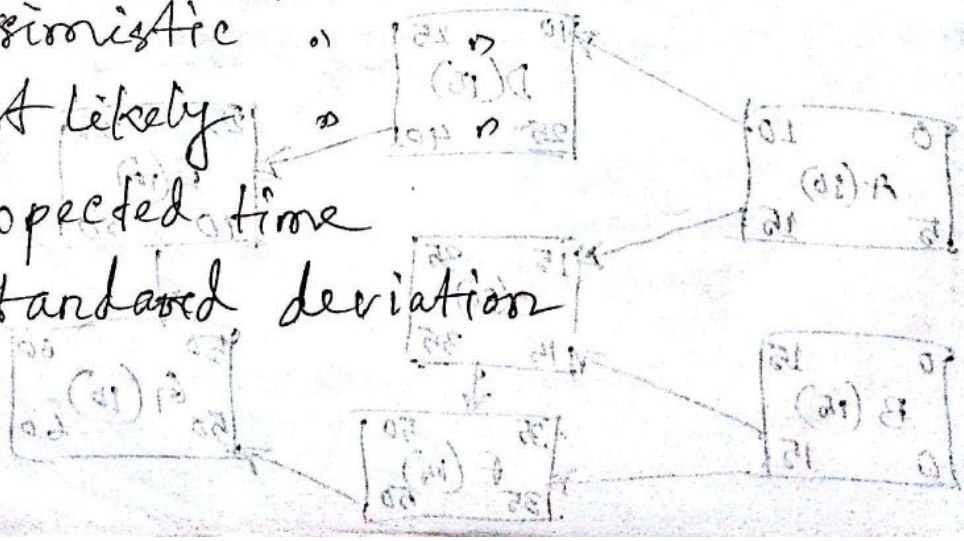
Slack time

Slack time = Late start - early start

* There is no slack time of the tasks of the critical path.

PERT

a = Optimistic time estimate
b = Pessimistic "
m = most likely "
te = Expected time
S_i = standard deviation



Expected time, $t_e = \frac{a+4m+b}{6}$

standard deviation, $s_i = \frac{b_i - a_i}{6}$

standard deviation of critical path,

$$s_{cp} = \sqrt{s_1^2 + s_2^2 + \cdots + s_n^2}$$

Example 1

$A = \begin{cases} m = 10 \\ a = 9 \\ b = 12 \end{cases}$ $t_e = ?$ $s_{cp} = ?$ $s_A = ?$

Soln

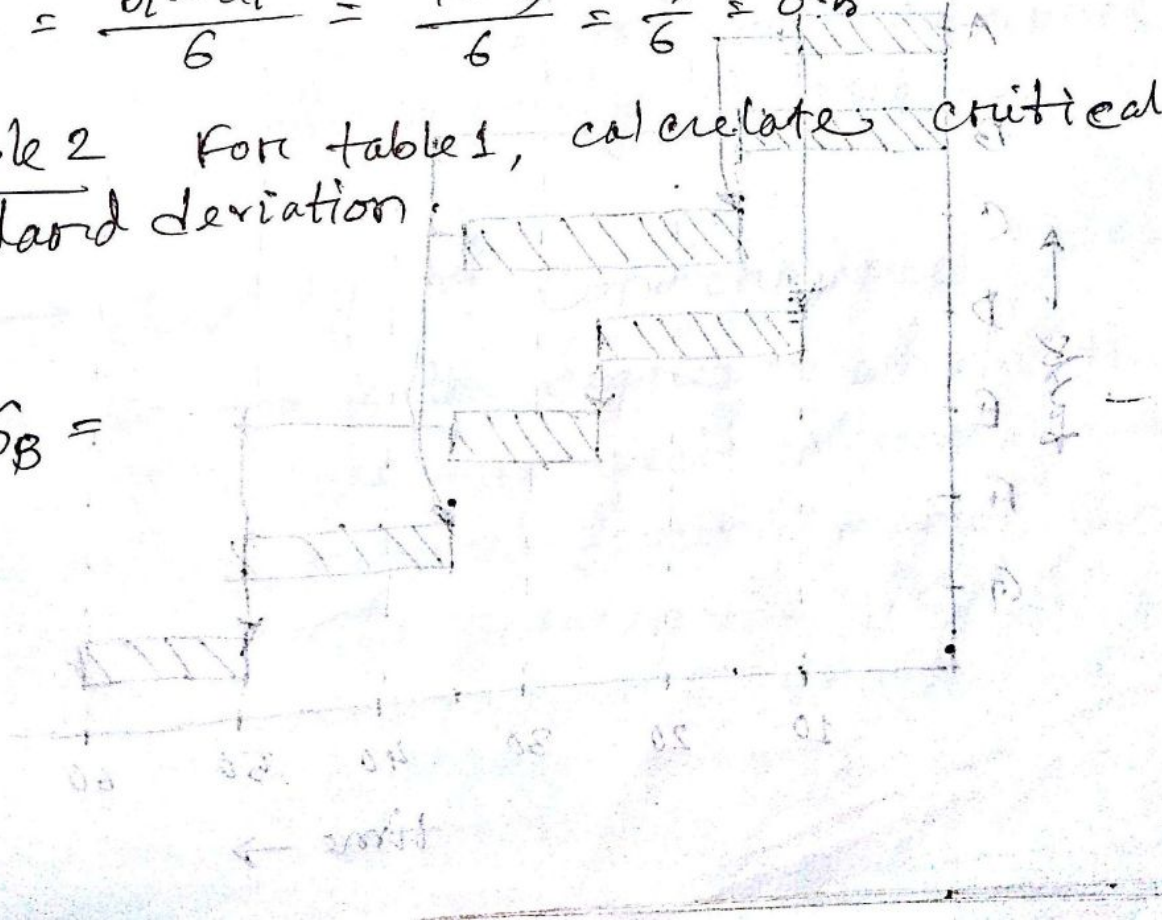
$$t_e = \frac{a+4m+b}{6} = \frac{9+4\times10+12}{6} = 10.167$$

$$s_A = \frac{b_i - a_i}{6} = \frac{12-9}{6} = \frac{3}{6} = 0.5$$

Example 2 For table 1, calculate critical path, standard deviation.

Soln

$s_B =$



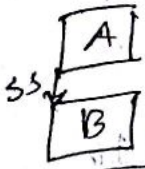
Bar chart

Gantt chart

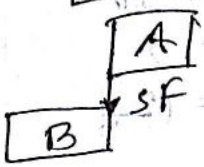
Dependency types →

1) Finish to start →

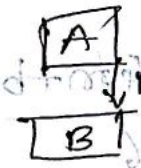
2) start to start →



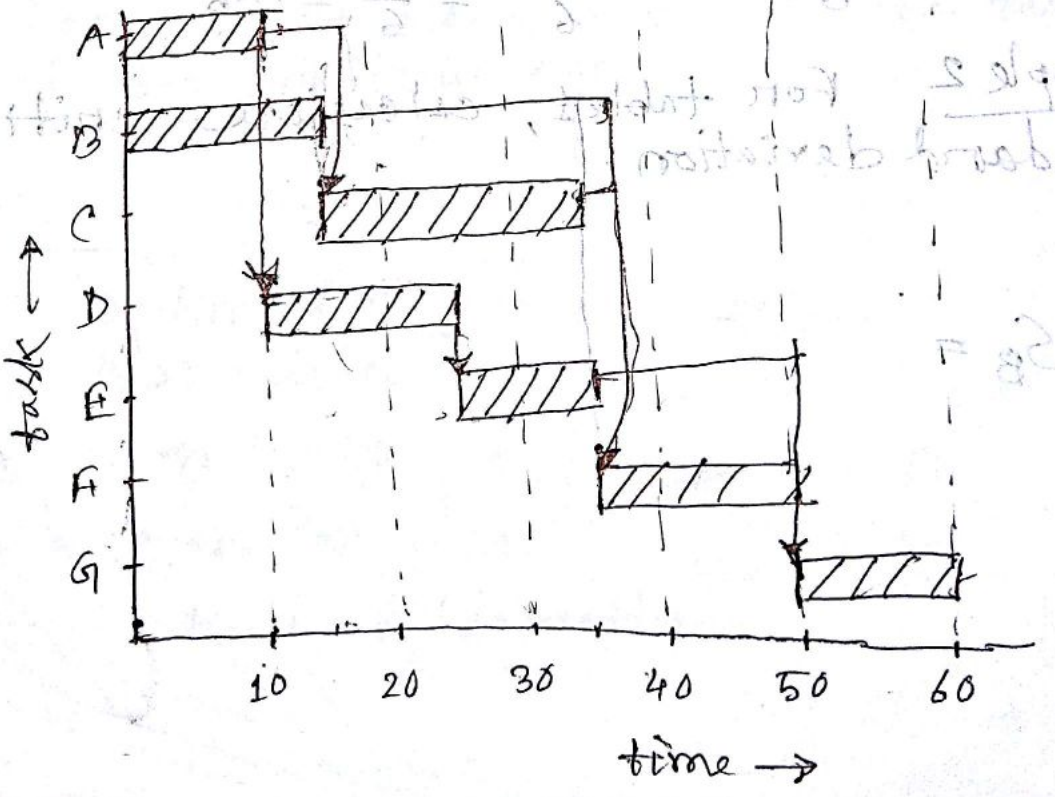
3) start to Finish →



4) Finish to Finish →



From table-1



Chapter-8

Quality

Quality refers to measurable characteristics - things we are able to compare to known standards such as length, color, electrical properties and malleability.

Two kinds of quality may be encountered -

→ Quality of design
  → refers to the characteristics that designers specify for an item. The grade of materials, tolerances, and performance specifications all contribute to the quality of design.

→ Quality of Conformance
  → is the degree to which the design specifications ~~att~~ are followed during manufacturing. The greater the degree of conformance, the higher is the level of quality of conformance.

User satisfaction = compliant product + good quality + delivery within budget and schedule.

Quality Control

→ Inspection
→ Reviews
→ Test

Quality Assurance

Quality assurance consists of the auditing and reporting functions of management.

Cost of Quality

The cost of quality includes all costs incurred in the pursuit of quality or in performing quality-related activities.

11(A) - Day

Date: 29/8/2017

Chapter-17

objectives

— Testing is a process of executing

White Box Testing / Glass Box Testing

→ কোডের প্রোগ্রামার ভিতরে কোথায় থাকলে

— Flow graph notation

→ First objective

→ Independent path কত তাকার।

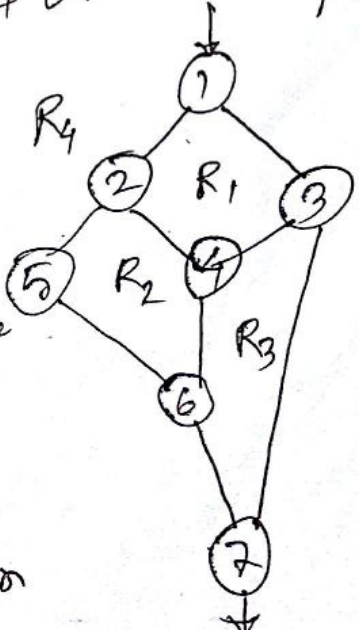
No. of path,

$V(G) = E - N + 2$

$= 9 - 7 + 2$

$= 4$

4 টি independent path থাকে



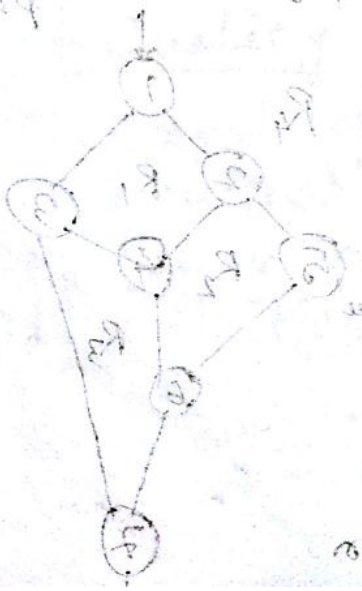
$V(G) = P + 1$ → no. of 2 out degree node

$= 3 + 1$

$V(G)$ = no. of closed region

Graph matrix

	1	2	3	4	5	6	7	
1		1	1					2−1
2				1	1			2−1
3				1			1	2−1
4						1		1−1
5						1		1−1
6							1	1−1
7	1	1						3+1
								=4



11(B) - Day

Date: 9/9/2018

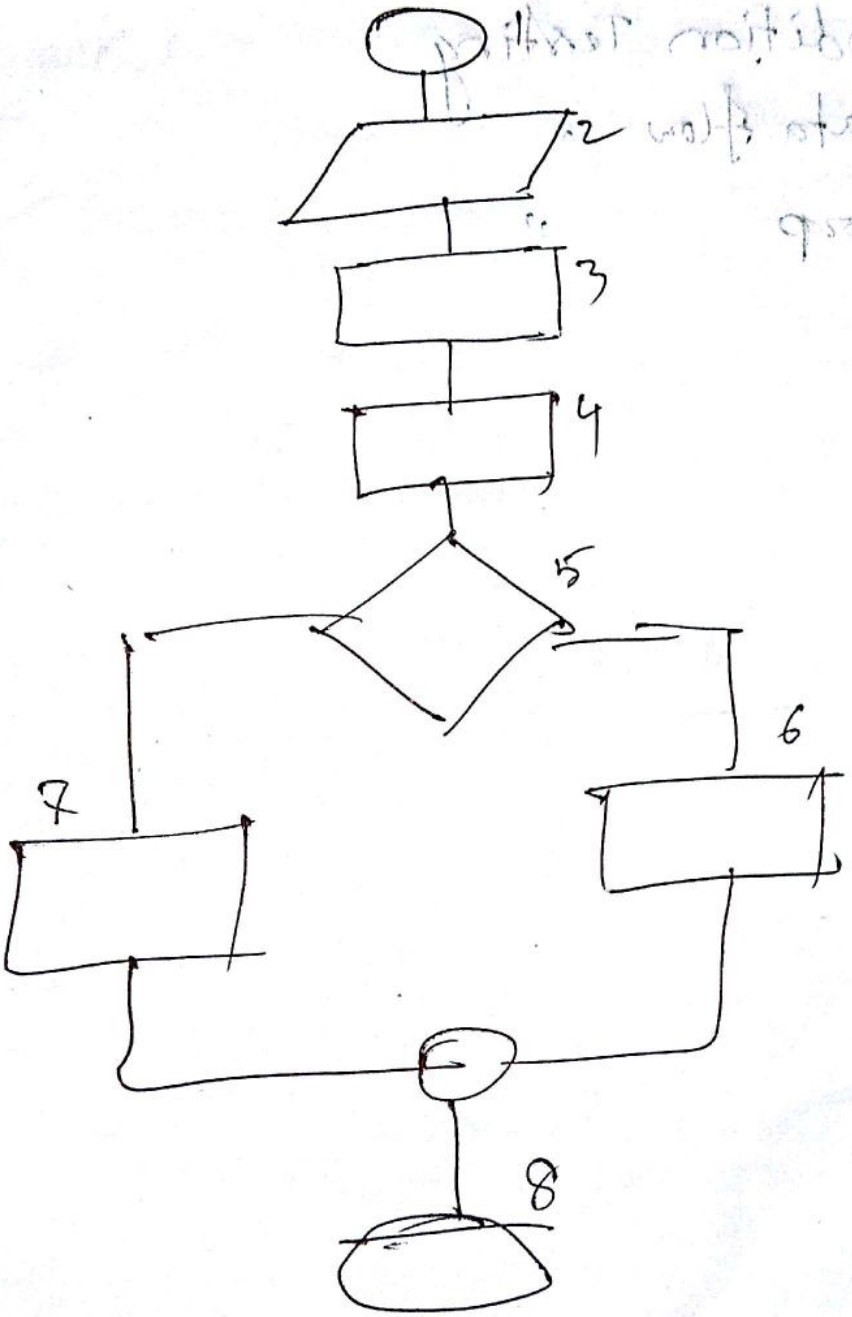
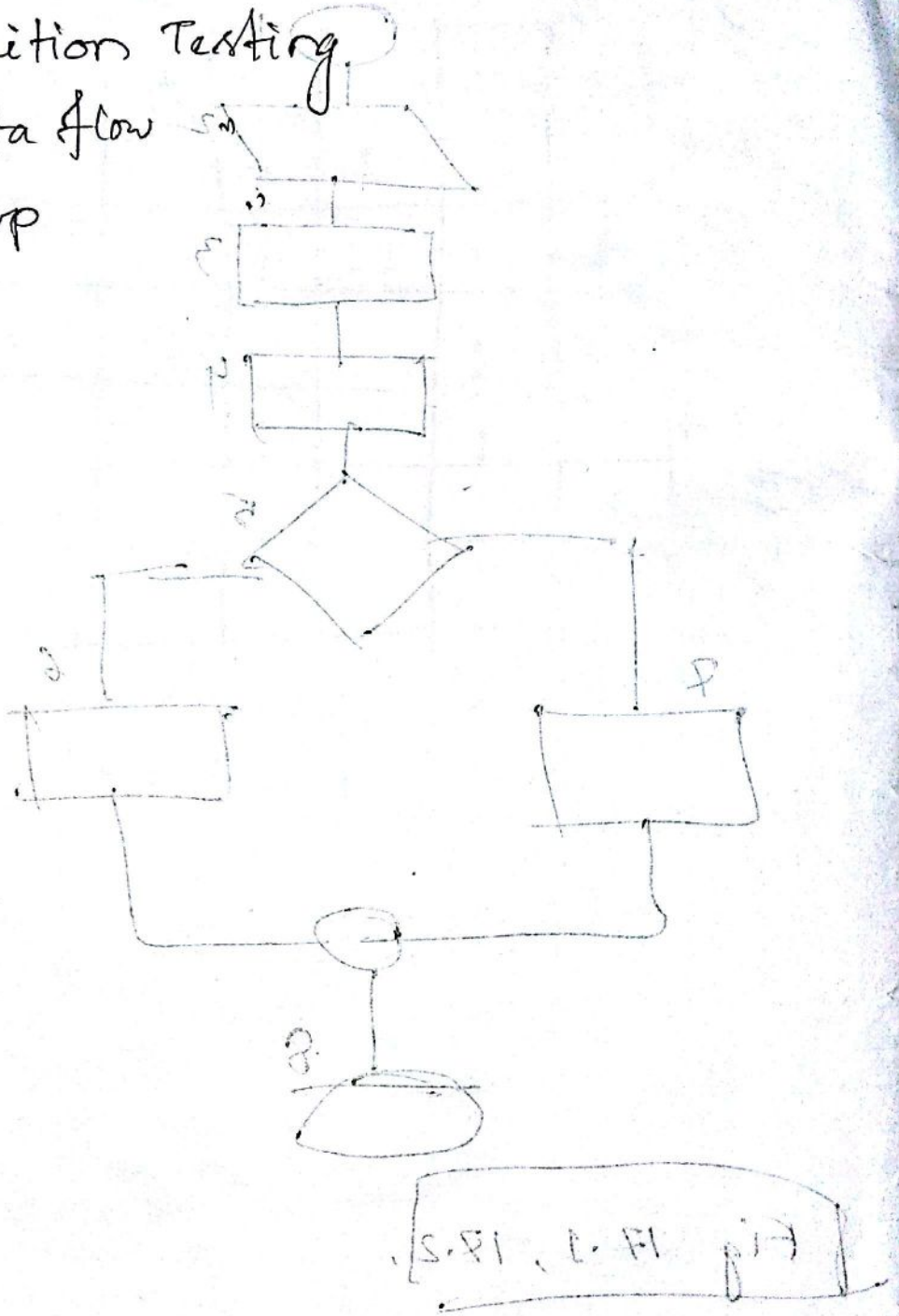


Fig 17.1, 17.2.

Control Structure Testing

1. Condition Testing
2. Data flow
3. Loop



11(E) - Day Date : 12/9/2017

Black Box Testing → Graph তৈরি করে characteristics খুঁজে বের করা হয়।

Category

→

Graph - Basd Testing

Chapter - 18