

→ 1st Apparatus
→ B was cancelled

(new topic)

Function Point Analysis

Refined 99

- software to estimate effort (effort) & price
 - user go function to quantity.
 - software development maintenance (M) measure
- ↳ independently use.

Lines of Code (LOC)

↳ for software to quantity 100% V/V.

→ 1 line to code

→ same code in 4 lines

↳ great programmes great 100% V/V.
like amount different 100%

Ex: windows vs mac

↳ so 2 software go price different,

but same code.

↳ 1000 LOC

↳ 1000 LOC

↳ So, function point go through to quantity.

according to requirements

swipe ← memory location

↳ So, price → low level usage of great code
depend on requirements → high in in code and

high in in code and

→ LOC go drawback, cannot measure 100% of the line of code 100%.

So, use function Point (FP) → more to quantity 100%
and also

FP Control Process

→ example Chu → input → output

Graphical view of standard functions

→ Data Functions → EIF (External Internal files)

→ External legal files

↳ OF TO (data use).

→ External interface files

→ report generate

→ form.

→ Transaction functions.

→ External Inputs.

→ External Outputs.

→ External Inputs.

Terms and definitions

Users identifiable → users (UT, GVE, MP)

→ both parties must understand

Control information → define

→ (UT, GVE) influence AP.

Elementary process → UT can make a change

→ users (UT, GVE) small part (UT)

Small part of process

(UT)

(GVE)

DET

unique, user recognizable, element of non repeatable table columns identifiable.

RET

diffuse & may relate to many objects.

(single table)
is a style
RET

related tables or greater refine recognize

ILF

App of RET code inclusion.

for - data objects save info (pic, date).

DET code as part regular, but

ILF complexity

Table w/ DET, RET (defined)

Average app find complexity, or

Table w/ many exam g.

find DET, RET, but T3C wrong

knows same, or

FP Trucker App

example.

→ 6 tables exist.

→ logical grouping. ~~of~~ → entity → T10

~~project id~~ → not DET (not user recognizable)

rest type, description → are all DET

So, primary key is DET

but, foreign key is DET

→ Table from scenario.

→ Process table ~~is~~ 8 ~~if~~ key, foreign key is DET.

but PK is not DET

→ Process DET and entity ~~is~~ group go next.

So, same count.

~~22/12~~ Today's PPT of IB
about DBMS
given by Prof. Dr. S. R. Rao
Date: 22/12/2013

Wanted only PK and
1 Q10, rest repeat
Topic contd all.

→ Process, Process DET, Process entity.

So, Process no. of RET = 3

→ 3 Q10 for PK & Q10, but FK will similar after, Q10.

1 Q10, count Q10 (Q10, Q10)
count Q10

FP

→ Q10 20 min → 20 min cost too much.

Q10 → cost + time Q10.

Problem We are asked to design an application -

Find ILF, AP

→ PK is student ID (assume).

Aus:

Student table

→ Student-ID (PK)

- name
- father's name
- mother's name
- DOB
- Gender

Courses

→ Course-Code (PK)

Course-id
Q10 (PK)

→ Student-ID (FK)

→ course name

→ semester

→ course-code

Full table
to open database
need PK.

uniquely identify
student ID number
→ get course code
for under 9 students

Sir sdn.

Student-ID (pri)
name
father's name
mother's name
DOB
gender

Course-ID (pri)
Student-ID (Foreign)
Course name
Course code
Semester

DET
(no PK)

ILF

DETs

RETs

Complexity

FP

Student

5

1

Low

7

Course

4

1

Low

7

14

Table given

RET

DET

1-19 g 20.

1

L

from given table

given table
row → 2,
so, last

multiple tables

2 tables from

→ RET → 2nd table ~~and~~ logically get it
first for 1st, they operate ~~get it~~,
they merge, and RET = 2.

1st FK
get count

2 tables
merge

(but ~~info~~ info

student-name¹ both table 9

after (not PK, not FK)

1st count both
2nd count both

only FK on T₂,

RET here ← { bin-foreign → count only }

student-name¹ bin-foreign ← count only

(both FK)

RET as rows 19 ← count only

RET as rows 19 rows 19

cols after 19 rows return

student 9 → prop rows 1 ← RET → last row

student 19 rows 19 ← RET

bin 99 rows 1

student rows 19
bin rows 19
student rows 19

29/05/23 (cont)

PMG1 Class 18 IS.

(last) → last for get recap.

→ FP → multiple tables → join on PK

→ FP → multiple tables → join on PK

→ FP → multiple tables → join on PK

→ FP → multiple tables → join on PK

→ FP → multiple tables → join on PK

FP tracker Application

example.

→ bad tables FWIW

1st table

→ Project table.

→ 2nd table FWIW

→ PK is project-id } → not DET.

Eduley table

→ project-id is FK here.

Press table

→ FK count as DET. (but I count)

PK does not count as DET.

Fusion Point ^{using} single det

bad
press
g
3 ET
merge,
so, RET
3 for
press.

RET → 1 since get are table.

DET → from each table.

→ one FP find

→ DET for press table
is FK double after
style count.

So, value different, according to the grouping.
 ↳ FP. { either and more change but }
 ↳ problem go into depends.

→ Grouping depends on scenario.

↳ merge and merge go result.

student

student

student

student

Problem

(example).

Student.

Student Course

Course

include
student

every student

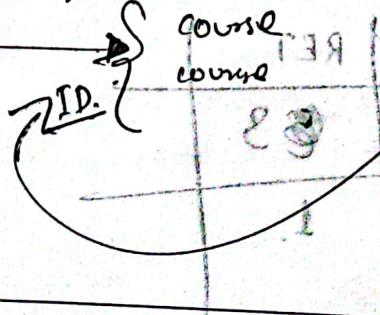
individual course.

depends on
student
course

course

(so, course description
here).

example class.



A table is given as follows;

total
4 T
table

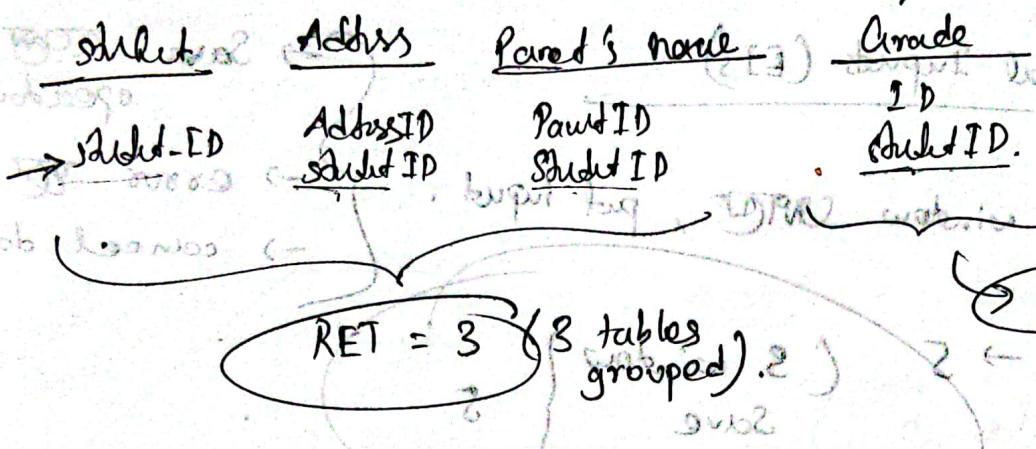
	<u>Student</u>	<u>Address</u>	<u>Parent's name</u>	<u>Grade</u>
--	----------------	----------------	----------------------	--------------

→ calculate the DET, RET.

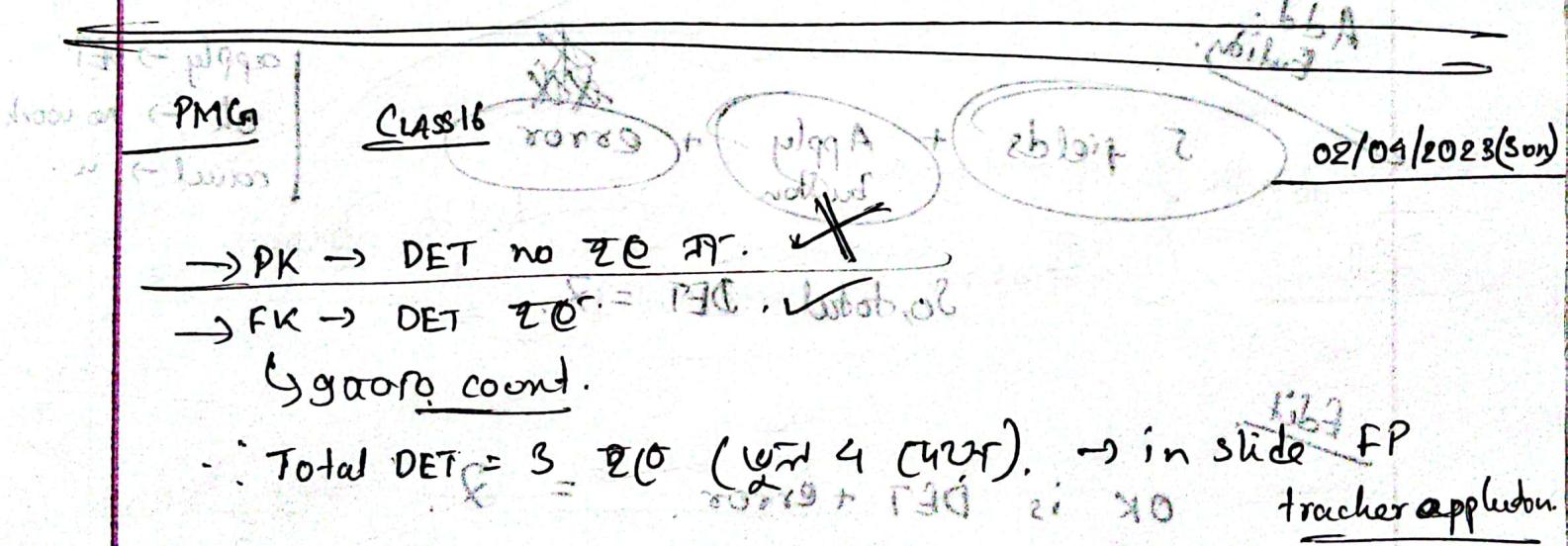
(Cptd)-

bid, grade with
 means CAPA
 1 student has
 1 CAPA, thus
 1 table
 can manage
 individual
 student
 course
 so
 one table
 for
 each
 student

Q 6 soln:



→ DET will depend on the fields.



① ILF → table names.

② RET

③ DET

④ Complexity.

⑤ FP.

1. \leftarrow subtbl (from given table).

2. Separate subtbls.

1st \leftarrow $\exists 0$ individual

2nd \leftarrow 3 $\exists 1$ table same, so group $\exists 0$

→ question 9 OC first. Ans 2008.

(new topic)

External Inputs (EIs)

Project → window CENTER, put input

L = EIS

other 1 hour DET → 5

Add
Eddy.

5 fields

Apply
button

error

apply → DET
OK → no error
cancel → n.

So, total DET = 8

Edit

update nice → (TUN) is now 0.8 = 2 DET total

other 1 hour

OK is DET + error

9.1

9.2

9.3

9.4

9.5

Deleted

Eddy delete → 1

Project delete 9.0 under 9 CENTER, so → 1

delete button → 1. let

Error → 1. file e has

Total 4

DET.

total 8 DET. 9.1 to 9.5 e ready up

handshake
21/92
P2A
detected

reject chain

reject valid

rejects

with probability

(1 - P2A) * P2B

(1 - P2A) * P2B * P2C

(1 - P2A) * P2B * P2C * P2D

Add Pass group

Apply + error

= 3.

(+R) subgroup

error P.

Dedede
Pass Group

A93

→ 2 debbles.

+ 1 staled
+ dedede button + error

So, total 4 DET.

Fib 7

shock

Add Pass

Fuelled so far 2. (with descriptions)

error + shock + debble

+ 2 1st debbles + error

So, fuel. ?

DET.

Dedede
if 3 debbles.

P2A total 2

3 debbles + dedede + error = 5 DET.

J = button

I = Fib 7

P = shock

so T2D total

class.

only
for
delete

node: ~~average~~

table go under

\rightarrow ~~average~~

FK in this table

here, 1 FK (s-id)

so, 1 table + grade table itself + delete + error button

student
ID is
FK of
one-to-one

Example (Sir).

Table

Student. F Name.

Student L Name

Student Address.

Create

Edit

C.P.A

Delete

A - A - 4

A - B - 3

A - C - 2

Create 9 DET

8 fields + create + error
button

5 total DET.

Edit 93 S.

4 total
DET

\Rightarrow Total DET of Create = 5
n n n Edit = 5
n n n Delete = 4

SRS

Date: 9/4/23

↳ basic info of the software needs to be explained.

→ user req., func.

→ non-functional reqs → software needs to be open 24/7.
 example: example

SRS Outline

1. Purpose → document the purpose explaining

→ target of the software

→ Future growth development, target market

info req'd

→ use case, functional requirements, non-functional requirements

2. Scope → get document which covers all requirements.

→ complex software can be very large, scope

subject will cover all the reqs from the system to the environment.

area of cover.

3. System Overview → similar to vision by scope

4. References → SRS document based on article, document, etc.

5. Definitions → SRS defines technologies, terms, etc. used in the system.

↳ system growth part + explain, for system to understand.

↳ non-technical people understand.

Use Case

↳ user system go, user facing interface
action, what define.

~~↳ software how work, not described in use case~~

↳ only user interface go, user facing interface,
what do.

~~↳ no internal working, no how implemented~~

↳ only says to action for action

button here, click action will do this.

Use Case template (8 WYSI)

→ use case first must follow this template

Name → unique name (name and number)

Summary → brief description. (for action use case).

Rationale → why this use case to action. → description.

Software.

User → use case, otherwise.

Admin panel, user panel.

use case to test 80%
70% (all users to put all use

use case go
Pre conditions → प्री कॉन्डिशन वा प्री कॉन्डिशन softwre
go condition.

→ Basic cause of events → गो ओर गो ऑफ -
→ serves of events.

Alternative Path → similar type of options,
BCOE goes to but
closely related go states
OTIS use case ने OTIS same use case ने under
Alternative paths.

Post Conditions → basic cause of events. completed
प्री कॉन्डिशन वा प्री कॉन्डिशन वा प्री कॉन्डिशन

BCOE
user action एवं
softwre resp.

action → respond.

steps of basic cause of events.

sensor receives info from output two

sub → Use case for Search and replace function. (कॉपी)

Name. - UC - S - Search.

Security →

to maintain security F.A.T, E.S, L, Q.A.

Rationale → text mainly replace is time consuming.

big file → So, need run frequently OT GAOYO search
and replace.

Users → All users (User's use)

Precondition → current load $\neq 0$.

BCOE → get (in points)

1. User wants
2. Software resp., asks for load
3. Gives load user \rightarrow and says all occurs reply
4. Software replies all.

and last 9 \rightarrow file \rightarrow desired output \rightarrow 10

Alternative Policy (points).

1. In Step 3, user says ~~not all~~ only the first \rightarrow software only 1st occurs reply
2. In Step 3, user says only search, no reply. So software does only highlight, end of use case
3. User can abort function at any time \rightarrow

Step 1, 2, 3, software return to \rightarrow precondition side

Post Cardon. → all objects replaced
by their replaced class.

→ examining use case frame

→ in book too, plus slide uploaded

↳ use case.

(reading).

→ CT this Wednesday.

↳ DFT, RET, ILF (calculator, table ---).

↳ Use Case

↳ SRS.

website link will be given

message with answer with att. file

message: answer, question, file
with regard to question

message with your own stored file: jmsisig33

. answer or correct possibility

: A. J. I. D. O. R. (J)

to learning of the material interpretation

reco to

(A. J. I. D. O. R.)

PROJECT MANAGEMENT Class 21

→ Non functional requirements:

→ Functional Requirements:

↳ Name, Summary, Rationale, Requirements, References.

→ Non functional requirements:

↳ example: the software should support all type of browser.

↳ example: the maximum size of a database on disk.

Kinds of non functional:

↳ Availability: the time frame the system will be up. example: System maintenance break time

↳ Efficiency: elaborate how well the software utilizes scarce resource.

↳ Portability:

↳ Integrity: Maintain data privacy of a user.

NF Requests: Requests + 30 char info

under 500 ms.

200MHz

RAM 32 MB

Microsoft Windows 2000
and 50%
CPU load

Robust → System can tolerate 20,000 errors problems.

→ scenario based 9 question fig.

scenario 1

scenario 2

scenario 3

~~Next CT on~~

DET, RET, Use Case, Functional Requirement,
Non Functional Requirement, SRS.

non functional

→ C4SRA - 4U - 4S

on last class (l Date)

2000, 2000, 2000, class

→ outline class 2000, 2000.

Availability, Efficiency, Portability, Integrity, flexibility

A E P D R

Performance, Reusability, Robustness

P R R

↳ flexibility.

↳ Performance

↳ Reusability

↳ Robustness → error handle against system crash.

JFAPB
PRF

Non-functional requirement example:

- ↳ Name
- ↳ Summary
- ↳ Rationale
- ↳ Requirement.

- ↳ Reference.

PRAG

Class 22

50/04/2023

(new side)
Software

Practise
Improvement

(Slide 10)

→ Purposes of software engineering
methodologies

↳ Organization go software making, improve AOT.

↳ CMMI prus improve AOT possible, TCM, improve BSI AOT.

↳ Evolutionary → change TCO ATO with time

→ organization go problems solve increased.

↳ organization go problems continuous improved

↳ needs patience, broad consensus only the software engineer
does not have to submit reports ← responsibility

Models / Certificates

CMM, CMMI, ISO 9000, Six Sigma.

CMM (Capability Maturity Model)

↳ Carnegie Mellon University.

↳ productivity, cost, performance... → govt. facts & best practices

↳ process fit a mature org.

↳ gov't gov't proc' fit of org' but

one way go't (different)

company & go' purpose differ

→ 1990s of CMM.

→ the users of CMM gave feedback.

↳ CMMI from the feedback.

↳ improves quality both

both software & hardware

CMMI Capability Levels (determine organization performance)

Capability Level 0 → incomplete level.

→ need changes to solve the problems.

SPI in practice
means any a guideline
model or standard as a guideline
for assessing and
improving a system

Capacity level 1. → initial level.

challenges

→ in respect to the challenges they face they have developed some pro-

Capacity level 2 → defined level.

→ ~~asymetricals~~ always monitor their performance level go or not match ~~according~~ according fact.

→ challenges to simple plan go ~~not~~ not overrule ~~do~~ do.

→ ~~as~~ ~~as~~ totally defined plan ~~they~~, and they do work following this guess.

→ goal go ~~not~~ match ~~according~~ according fact, focused eff.

→ organization has mission, CMMI never.

→ improved version of CMMI.

CMMI has 5 levels Madridrads.

Madridry level 0 → incomp. level.

→ organization carry out ~~not~~ not

~~on time~~ ~~on time~~, ~~or it may or may not find~~

→ ~~as~~ finish when absolutely necessary.

Maturity level 1. → initial level

→ organization ~~at~~ purs/ ~~at~~ task

coupled ~~at~~, but deadline

go ~~no~~, ~~at~~ over budget ~~at~~

but still coupled

→ these organizations are unpredictable.

→ change ~~at~~ problem

Maturity level 2 → managed level

→ planned, executed, reviewed.

→ my not be related → they follow pros and coupled fact.
of orgn.

Maturity level 3 → defined level.

→ pros follow ~~at~~.

→ change ~~at~~ not proactive ~~at~~.

not worry.

→ change ~~at~~ ~~at~~ predict.

ML 4 → Quantitatively measured level.

global view → follow data driven.

of internal

all → controlled way! ~~at~~ handle

process

objectives → Quantitatively handle.

DEF, JET, ILF,
IRS, Fuel/Non fuel requirements
Use Case

CT syllabus → 6, 2, 8, 9 slides.

→ next Wednesday Quiz.

→ 15 or 20 marks
3 or 4 questions

MLS → Optimized level.

→ ~~fixing~~ press CT continuously improve.

→ choose correct solution.

→ quickly solve.

→ stable, work towards innovation.

CMMI.

Software engineering process group

↳ solve problems.

several dozen KPA, e.g.
request trace, print plan etc.

→ KPA → key press

→ specific goals and tasks.

→ CMMI → what not what

→ each KPA → specific

→ organization of action taken to solve → SE procedures.

Assessment process

↳ local CT assesses ISO 9001 STS

↳ CT → assessor → CT

↳ 1 week delay, see documentation.

↳ assessment report → identifies CT processes follow

on demand of Madhuri level CT CT

practise CT missing ISO 9001 CT

↳ find ways to solve.

→ Head and sub of KPA.

→ Print plan KPA v.t.)

a) establish

b) dev. a print plan

c) obtain conduct

to the process

CMU, ISO 9000 → Very important.

CT. ~~Final syllabus~~ → 6, 2, 8, 9 slides.

ISO 9000

- ↳ quality needed.
- ↳ customer go requirements to fulfill organization's fact.
- ↳ organization go performance to improve org.
- ↳ set of pass / fail standards.
- ↳ have certificates.

→ Assessor → go audit by vir.

- ↳ can determine what clys need.

(slide) 8 core principles:

1. Focus on customers' needs.
2. Leaders act, CT make sure org. fulfill.
3. At top level go org. is imp.
4. Quality need.
5. Interrelated process go and rev.
6. Continuously improve performance.
7. Well informed decision.
8. Organization and processes → TQM based quality.

Review

↳ current work putted to reviewer to get their

↳ feedback

on if work product ok or not.

↳ different wp to different reviewer.

Software to review develop review.
Project development.

↳ I review technique used for all wp,
different wp different review technique.

→ Main purpose of review → find defect in product

↳ fixes in product down

↳ stakeholder's approval.

↳ professional developed & help

↳ code review by CMMI,

→ wp after review has fewer defects.

→ review product to identify, analyze, solve,
problem/defect, review, decided plan

so later
no need go

~~Document review delivery~~

1) Inspection

↳ formal review technique.

↳ inspector (to) ~~not~~ CTO, take feedback, improve wp.

↳ test plan inspected, SRS.

↳ fix first approach by 6/15.

↳ depend ACU (automate) prob, stand one need, so, inspection helps

↳ inspection team, moderator, needy.

needy go after inspectors w/o CTO.

→ node due defects

→ inspection go main goal all defects fix repair for better up in future.

→ prevent defect.

→ serious defect fix dev. do not endue prudish, stand over, effort expense, team needs to backtrack.

- implemented is easy.
- immediate outcome is quickly.
- work for judge, so glasswo.
- nodes of work wasted ~~can~~ save us.
- need well-chosen team
 - moderator to run nicely.
 - wp go author/criticism & open mode
- Project user inspector dean & 3-10 people

↳ different perspective.

↳ buyer } use different documents.
 ↳ programmer

↳ first view requested, imp. for early de-

→ also choose a moderator (to run nicely).

↳ evaluate wp
 ↳ issues dev. inspects
 ↳ control nicely.
 ↳ open and ~~open~~ discussion
 ↳ stop and keep as

- ask
- The project manager should be an inspetor since he knows about project
 - Good moderator COULD GO TO MPC.

② Deskcheck.

↳ simple review technique.

↳ up to author gives the copy of the product to reviewers.

↳ team members find the defects and send to author.

↳ vision & scope demand, discussion summary.

↳ programmer code first wants review, without do others for checking → get feedback.
↳ improve code.

→ for some products, no need huge team.

→ Deskcheck has no written back.

↳ no follow up meeting.

↳ not a formal review.

↳ no standard.

↳ reviews
simply review
and read.

- no moderators.
- no ~~goals~~ generate
- lack of formality
- project notes with desk check.

using self

- Deskcheck can be used before

inspektion

Procedure to inspect

author send to peer → feedback
uniformly improve the prod.

ex. after deskcheck 9 errors
avg 210

③ Walkthrough

informal way of presenting technical doc.

↳ author fit walkthrough run, meedy call, TPA, invite reviewer, take comments, make sure all present understood.

- author presents in front of audience.
- slide presentation, with well^{high} details.
- dyn presentation & well^{high} is a common review technique.
- well^{high} go w/ ~~most~~^{some} ~~modern~~^{student} materials (UT pushed ~~at~~^{to} students) (UT ~~is~~^{is} good)
- well^{high} is a ~~obstacle~~^{brainstorm} to my technique.
(~~not~~^{my} idea)
- author go ~~to~~^{the} detail parts ~~of~~^{of} a clearly presented ~~out~~^{out}

Audience of well^{high}

- ① All receivers (users, stakeholders, ~~major~~^{major} least) all present.
- ② UT gave ~~are~~ and ~~force~~ ~~pushed~~ gave ~~up~~
- ③ All sections of material present.
- ④ ~~gave~~^{to} ~~pushed~~ UT gave ~~up~~
- ⑤ Missy section UT present.
- ⑥ At least ~~one~~^{one} ~~part~~^{part} ~~and~~^{and} ~~but~~^{but} in

- ⑥ After ready author ask for input
- ⑦ Issues need to be ~~BTI~~ T discussed

⑧ Code Review

- ↳ team code sample → examine ↗
↳ bug fix the defects.
- ↳ given block of code → incorrect, can improve
↳ given code to ~~BTI~~ imp. code review.
- ↳ bug finding, bug fixing → helps.
↳ cross training. → junior dev can learn new things
→ good, effective communication

~~take 10s.
works do do~~

- ① Sample code just do review

(full review is quite impossible)

400 line of code (Java) review almost 2 hours
need high level line.

↳ depends on team, complexity of code

- ② Tricky or complex code to review →
single code A review →
→



Rule of Thumbs for CR review.

- ① Is there any one expertise in code?
→ good for code.

→ rest of the documents form →
↳ very → after, so good.

- ② Difficult or abstract code →
→ more error-prone.

- ③ Some objects, library is difficult to work with.
more difficult if non-intrinsic interface →
error-prone.

ODA: i-
Gia

④ Code written by inexperienced?

→ GUI guyne de hoga?

Bugachar large function?

↳ if so, then error prone.

⑤ Some code area to problem to catastrophic.

↳ Jax screen is more catastrophic than splash screen
written

↳ so, review none carefully.

CR Checklist

① Cleanly → easy to understand

→ MFBY

→ refactoring

② Maintainability → GUI guyne se koi nahi

→ well documented.

③ Accuracy → code does what it is supposed to do
→ algo w/ a implemented?

④ Reliability and robustness → fault tolerance
→ error tolerant.

→ unexpected condition
exceed me?

⑤ Security → modification & vulnerable frag.
→ well tested & as it is

⑥ Scalability → code runs fast if
increased load, code handles more
at once fast.

⑦ Reusability → can this code be used in
other target application

⑧ Efficiency → CPU cycle, bandwidth, resources
efficient use fast, ok?
→ cost for data update?

CR go part is Pair Programming.

2 programmers work on 1 style code.

→ part of XP (extreme programming)

→ 2 ~~at~~ review and code.

→ helps the professionals do (get) better
work (code) faster.

→ Describe question from here. (This topic)

→ code review is specific for Code reviews all.

→ we go ~~at~~ different

→ Document review

→ So, different, cannot tell which is better.

COCOMO Model.

- Construction Cost Estimation Model.
- 1981., Boehm propose AOCU.
- software factors \rightarrow cost factors afford schedule } determine
- (KLOC) → size of project by kilo of lines of code for estime.

Types of project.

Organic: Well understood.
→ team size small, experienced.
e.g.: simple business system.
subdivide system subsystem.

Size: (0 - 50) KLOC.

less than 50,000 lines of code.

② Semi-decked: → some exposed stuff
→ n inexperienced n.

→ some OOP, some unsafe

e.g. → flyby system

dedic system
coupled system

size (80 - 300) KLOC
sue.

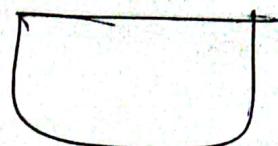
③ Embedded: hardware strongly coupled with software

e.g. ATM, air traffic control,

sue. (800 KLOC)

Basic COCOMO Model.

table given



$$\rightarrow \underline{\text{Effort}} : a_b (KLOC)^{b_b}$$

↓
total effort
↓ dev.
↓ dev. sign.

↓
pushed.

unit is person/month

$$\rightarrow \underline{D} : C_b (\text{Effort})^{d_b}$$

↓
↓
effort
↓
time to
dev. sign.
↓
unit (month)

→ KLOC and co-efficients go at 65% can find.

$$\rightarrow \underline{SS / P} = \frac{\text{Effort}}{D_{dev}} \times \frac{KLOC}{PM}$$

↓
staff/
no. of persons

↓
time to
dev.

~~1/10e.~~

Example 1.

so, effort ad

dev. due area CV

Aus: eqn. $CV = \frac{1}{2} \times 90 \times ab \times \text{eff}$
 $\Rightarrow \text{eff} = \frac{2 \times \text{Aus}}{ab \times 90}$

Example 2

→ determine first 9 pycd \overline{D}
CV type.

$\rightarrow (50 - 300) \text{ CV semi double}$

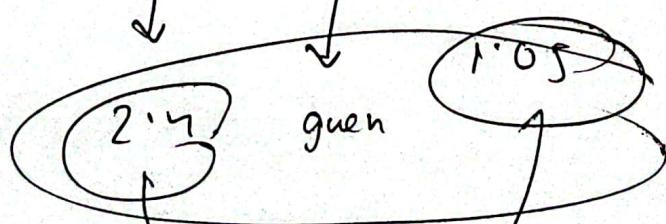
semi fully exposed 3 \overline{D} → (p b)

PY 218

Slic

Organic \overline{D}

$$\text{Effort} = ab (\text{KLOC})$$



(COCOMO) Model and COCOMO model

→ COCOMO Model questions → find ~~CAT~~ type
by ~~function~~
cycle / size based/
cyclic.

→ find by ~~function~~ → size of code with respect to complexity

→ COCOMO model ~~with~~ I ~~for~~ question ~~at~~ ~~at~~

→ table will be given in question

size of code = $\frac{C}{M}$ \times $\frac{F}{P}$ \times $\frac{D}{R}$ \times $\frac{S}{T}$ \times $\frac{E}{A}$ \times $\frac{G}{B}$ \times $\frac{H}{C}$ \times $\frac{I}{D}$ \times $\frac{J}{E}$ \times $\frac{K}{F}$ \times $\frac{L}{G}$ \times $\frac{M}{H}$ \times $\frac{N}{I}$ \times $\frac{O}{J}$ \times $\frac{P}{K}$ \times $\frac{Q}{L}$ \times $\frac{R}{M}$ \times $\frac{S}{N}$ \times $\frac{T}{O}$ \times $\frac{U}{P}$ \times $\frac{V}{Q}$ \times $\frac{W}{R}$ \times $\frac{X}{S}$ \times $\frac{Y}{T}$ \times $\frac{Z}{U}$

or $size = C \times M \times F \times D \times S \times T \times E \times A \times G \times B \times H \times C \times I \times J \times K \times L \times N \times O \times P \times Q \times R \times M \times S \times U \times V \times W \times X \times Y \times Z$

most of variables will be constant

(John McCall's factors) fact 1 & 2 ←

form of size = $C \times M \times F \times D \times S \times T \times P \times N \times Q \times R \times O \times V \times W \times X \times Y \times Z$

size of code = $C \times M \times F \times D \times S \times T \times P \times N \times Q \times R \times O \times V \times W \times X \times Y \times Z$

size of code = $C \times M \times F \times D \times S \times T \times P \times N \times Q \times R \times O \times V \times W \times X \times Y \times Z$

size of code = $C \times M \times F \times D \times S \times T \times P \times N \times Q \times R \times O \times V \times W \times X \times Y \times Z$

John McCall's 12 factors → Knott Model

and for programming

and estimation

size of code = $C \times M \times F \times D \times S \times T \times P \times N \times Q \times R \times O \times V \times W \times X \times Y \times Z$

Gantt chart.

↳ project go timeline go visual representation.

left go all tasks

Sync timeline

↳ can be described

→ first go left, then first go sync to QC,

→ first sync then sync.

Time Estimation for tasks

→ 5 in task (considering waterfall model)

→ depending is go comb in sync go sync to next

→ Planning comb sync sync.

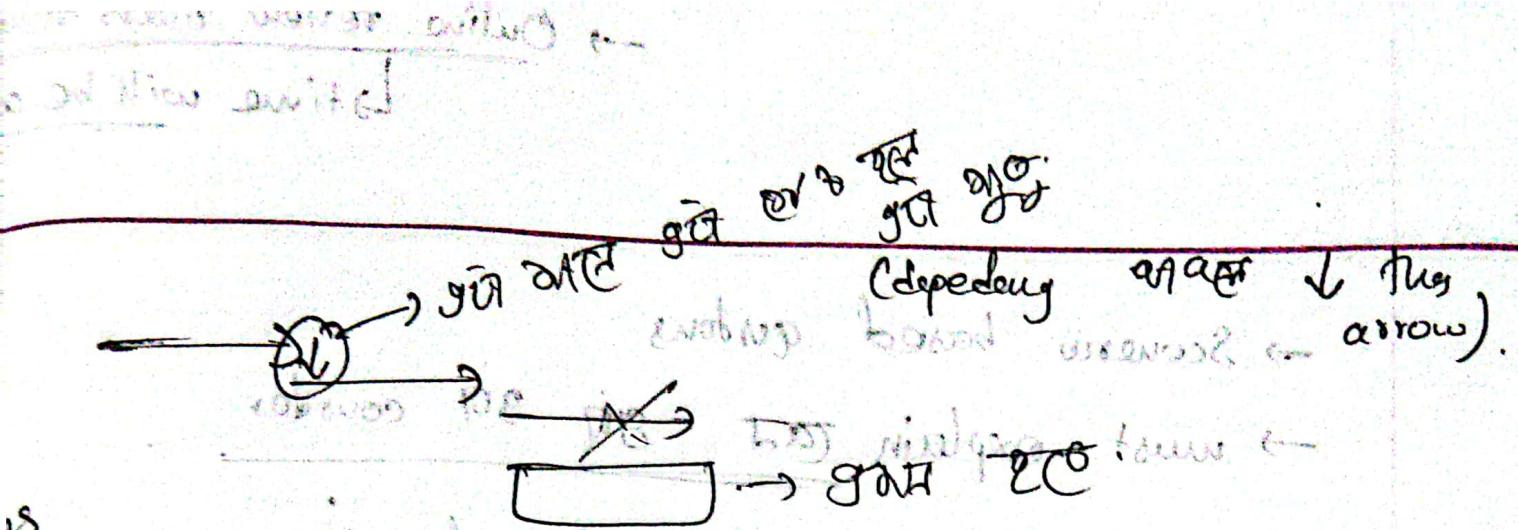
→ Body go sync body,

→ Deployed → deploy go sync

Gantt chart. → this example is waterfall model,
question may not have answer

→ horizontal lines.

→ bar sync represented using bars below each other

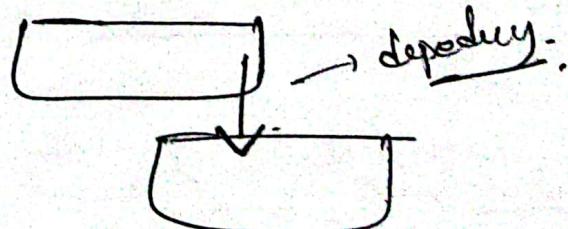


- ① Task identifying (must be logical).
- ②先后 priority
- ③ Gantt chart
- ④ dependency starts ↓ the arrow.
- ⑤ Task work now knows timing.

→ time goes by depends on scenario.

- I will give you consider days.
- here example in weeks.
- must be logical.

Note: We will give



rectangles, not arrow

→ Online review class tomorrow

↳ time will be announced

→ Scenario based questions

→ must explain what you consider

→ Cannot draw question only fig

Assignment will be uploaded.

↳ Online submission → 50% publ. ②

↳ cannot draw → 25% ③

↳ submission exam dy.

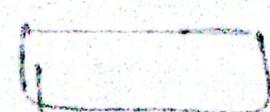
otherwise no grades MFT DE emit ←

→ publ. submit 50% fig & B16 1 ←

→ draw in alge book ←

→ logical or true ←

↳ publ. ↴



→ fig. Max size : of all

→ can have multiple

PMG

Online Class (cont.)

5/4/23 (Thu)

Ground & Chart

→ first by task and see dependency.

Parse will be given

Step 1 → task define (from parse).
→ duration assume.
→ dependency identify (task A → task B, task C → task D)

↳ task A ends at task B cannot start so dependency
↳ understand from scenario;

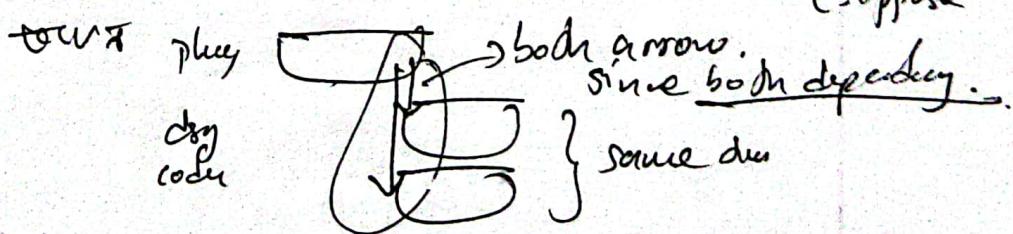
Step 2 Draw duration by rectangle.

→ final time depends on total time (here week),
but not the day.

→ can start multiple at the same time.

task if code and dry both after play.

(Suppose



→ dependency → after arrow fire to it.

→ Assumed g explain (in great detail)
is just g fine \Rightarrow for g fish.
(explain detail)

Question pattern

→ 5 marks \Rightarrow 8 tf.

$$5 \times 8$$

→ Slide 6 to 13

→ DET, RET, Use Case, FR, NFR, SRS,

Review, COCOMO Model.