

UI design

17/06/23

UI design



user → interface → service → interface

UI → output → screen → text → image

Input → keyboard, mouse, voice → software system → consideration

↳ sensor input

→ easy to operate → fast response time

→ requires more UI design work from developer

UI → req. analysis → 3 part (a) → gives

Usability

UI must be usable by the user

UI → 3 parts → error detection

↳ erroneous input UI → detect more →

Prototyping

- not a real implementation
- scaled down implementation
- incomplete version
- UI നി ഒരു പ്രോട്ടോട്ടേജ് അഥവാ പ്രോട്ടോ
- first version of UI
- user റെബുക്സ് കുറഞ്ഞത്
- user ഓൺ ഫീല്ഡ് ഓഫ് മെറ്റിംഗ് എന്ന് അറിയപ്പെടുന്നത്
- user requirement ഓഫ് അഭ്യ അഥവാ ദിസൈൻ വിവരങ്ങൾ
- user requirement ഓഫ് അഭ്യ ലഭ്യ മാറ്റ്
- requirement check എന്ന് ടെസ്റ്റ് കേസ് എന്ന് അറിയപ്പെടുന്നത്
- coding ഓഫ് ടെസ്റ്റ് കേസ് ഗെനേറേഷൻ ഓഫ് ടെസ്റ്റ്

e.g. sorting routine test

→ random case test

best case, worst case test

→ data unsorted sorted എന്ന് കിട്ടുന്നത്

→ അവ ഓഫ് സോർട്ട് എന്ന് അറിയപ്പെടുന്നു

test case document

(UAT) Test Case Report

→ UI design

→ system requirement

team building

→ लोग UI वा डिज़िटल मार्केटिंग

करने की आवश्यकता है।

Methods

→ Code prototyping

→ UI builders

→ VS } button, label, text field etc.

→ MS Front End

→ Paper prototyping

→ ताज़ा चिन्ह (sketch)

→ लाम्बाएँ जीवन ग्रन्थ

→ at a glance इसे निखारी देखने वाले
(more visual bandwidth)

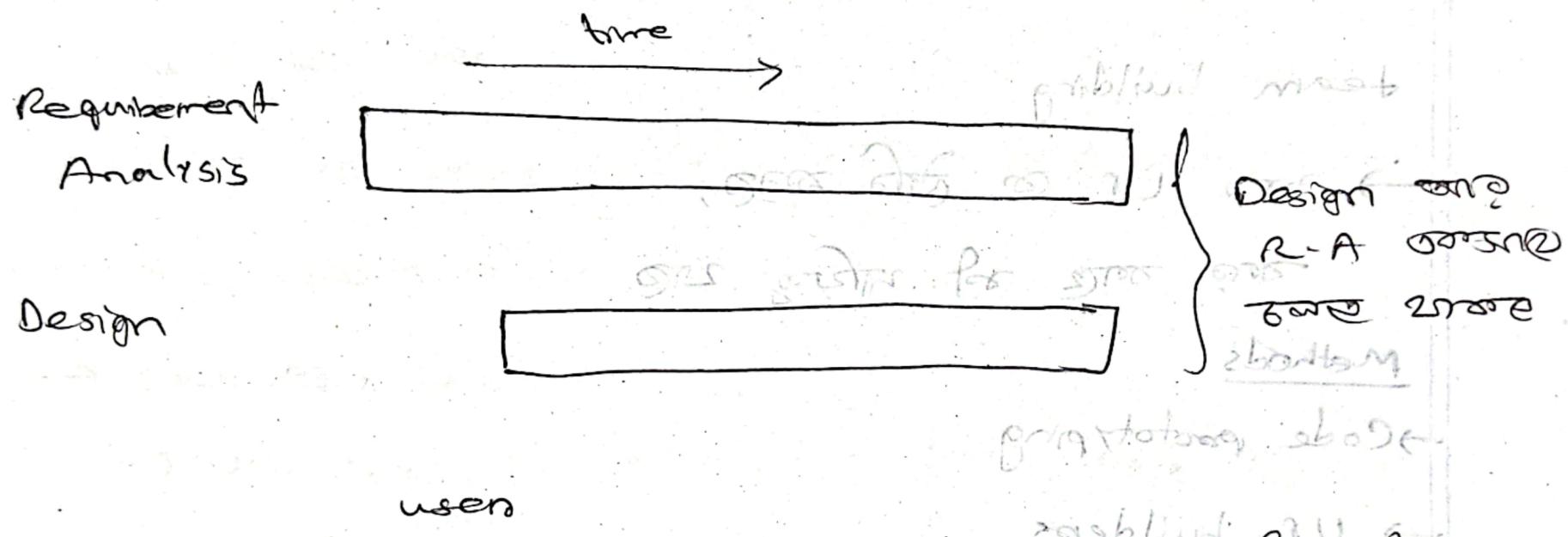
→ very easy to change (just erase and re-sketch)

→ non-technical → users से उत्तम व्यवहार

→ users को feel नहीं करते हैं ऐसे गले → उनके uneasy
feel को modify करते

Rational Unified Process (RUP)

→ represents any process model



- prototyping ~~requirement~~ requirement or ~~user~~ user furnish ~~user~~ user
- users ~~to~~ goal ~~to~~ user ~~to~~ satisfy ~~user~~ user ~~user~~ user
of check ~~user~~

UI design considerations

1. Stroke for Consistency

→ design consistent

একই কিনি হৃতি রয়েছে তা

ক্ষেত্রে এ রয়েছে প্রতিবন্ধ করা

e.g. files input অথবা স্লাইড অথবা ফিল্ড

ক্ষেত্রে screen তে এর আপোনায় same enter

2. frequent user ক্ষেত্রে keyboard shortcut দেওয়া

→ efficiency

→ fixed / configurable

3. Informative feedback

→ destructive / irreversible operation ক্ষেত্রে prompt দেওয়া,
by default NO select

4. one step এক টে আলোর step এ ছাই, তাহি অনিয়ে গুচ্ছ

→ user ক্ষেত্রে প্রয়োগ সাফ্টেণ্ট ক্ষেত্রে গুচ্ছ

→ closure এক টে মুছি তাহি identify ক্ষেত্রে dialog

→ තුළ info දෙනාගැනී නො යොමු කළ ලද රුකුණු button

⇒ මෙයක පෝපු-ඩූප් නො යොමු කළ ලද ලද ලද

→ නො නිවැරදි නො නිවැරදි නො නිවැරදි නො

5. Simple errors handling

6. Easily නො නිවැරදි නො නිවැරදි

7. user නො නිවැරදි නිවැරදි නිවැරදි

8. users නො නිවැරදි නිවැරදි නිවැරදි

e.g. dashboard

SAP → #1 SW company in Germany

↳ extremely configurable

→ difficult to operate

18/06/23

UI design components

tool box

button text → action → verb ~~more~~ ^{more} ~~less~~

tool bars → common actions ~~more~~ ^{more} ~~less~~ probable

menu → infrequent action ~~more~~ ^{more} ~~less~~ ~~more~~

button → frequent basis ~~more~~ ^{more} ~~less~~ operation

tool bars → always more?

list box → contains many elements ~~more~~ ^{more} ~~less~~ ~~more~~

list box → element ~~more~~ ^{more} ~~less~~ sorted ~~more~~

→ search box

combo box → compact ~~more~~ ^{more} ~~less~~

→ ~~more~~ ^{more} ~~less~~ ~~more~~ ~~more~~

tabbed pane → ~~more~~ ^{more} ~~less~~ ~~more~~ ~~more~~ screen

checkboxes → ~~more~~ ^{more} ~~less~~ switch ~~more~~ ^{more} ~~less~~

dialog box → temporary ~~more~~ ^{more} ~~less~~ ~~more~~ ~~less~~

CT syllabus: UI design

scenario ~~more~~ ^{more} ~~less~~ → ~~more~~ ^{more} ~~less~~ UI design

Use case

- requirement analysis තුළ part → s/w and bst
- සංස්කරණ න්‍යා මත requirement නේ නිශ්චිත ප්‍රාග්ධනය ඇති න්‍යා මත requirement නේ නිශ්චිත ප්‍රාග්ධනය ඇති න්‍යා මත
- Actor → s/w හා න්‍යා මත users න්‍යා මත programs → s/w and bst
- actors හා න්‍යා මත → goal න්‍යා මත
- goal හා execute න්‍යා මත සඳහා න්‍යා මත න්‍යා මත → steps
→ use case
- requirement හා bundle න්‍යා මත express → දෙක් s/w and bst න්‍යා මත න්‍යා මත න්‍යා මත න්‍යා මත න්‍යා මත
particulars part හා represent
- Use Case Diagram
- traceability matrix න්‍යා මත
- requirement to use case න්‍යා මත
- traceability matrix න්‍යා මත → many bstdy
- න්‍යා මත use case න්‍යා මත න්‍යා මත න්‍යා මත
- traceability matrix න්‍යා මත determine
- infuse to s/w and bst → requirement to use case

~~Defining~~ Use case

moorish 2013

Identifying

mpab 10

dictionary attack → all possible keys for access to db

mpab ← moorish 24/06/23

participating actors → system (mostly)

superset mpab
system-to-be → target system
methods not
→ database
switch
lock device
servers

initiating actors
human being ←
participating actors

offstage actors → record keeping by db

03/07/23

Class Diagram

Requirement gathering
UR design

} Req. gathering

Use case writing → requirement analysis

Class diagram → design

Design (Object oriented design pattern)
class
method
attribute
class hierarchy
interface
association
aggregation
composition
generalization
specialization
design techniques
for problem solving

→ ගෙන්රිජ්‍යා ස්ට්‍රේම් නෑතුව පැවත්වනු ලබයි

TE වෛද්‍ය පෙනී තු මෙහෙයුම් පිළිගියුරු කිරීම් → fixed model

→ ප්‍රකල් හිමිත් ගෙන්රිජ්‍යා ස්ට්‍රේම්

object oriented design → object oriented

08/07/23

Interaction Diagram

→ class ↔ class ↔ interaction

Sequence Diagram

Collaboration Diagram

Modelling

→ change / modification becomes easier

→ modification less complex

* → represents multiplicity

Modeling → অন্য অবস্থার মধ্যে পরিবর্তন

time | → time ↔ direction of time sequence
(GR) ২১৩ → কলাম করে সেগ রেখা
করো

vertical bar | ⇒ object instance ↔ lifetime

message → thick line → polymorphism

return → broken line (return)

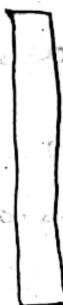
→ partial message

↳ parallel → parallel message

asynchronous call



synchronous call (a return value)



return value is always local

→ അനുഭവം കൊണ്ട് നിരക്ക് പില്ലാൻ
അനുഭവം കൊണ്ട് നിരക്ക് പില്ലാൻ

→ എടുത്ത ദാഹനം കൊണ്ട് പില്ലാൻ
അനുഭവം കൊണ്ട് പില്ലാൻ

→ എടുത്ത ദാഹനം കൊണ്ട് പില്ലാൻ
അനുഭവം കൊണ്ട് പില്ലാൻ

Success case failure case

→ കമ്മറ്റി ആശീർവ്വാദം കൊണ്ട് പില്ലാൻ
→ കമ്മറ്റി ആശീർവ്വാദം കൊണ്ട് പില്ലാൻ

Collaboration Diagram

→ lifeline 2020 08/07/23

08/07/23

Instance scope → getter, setter methods

↳ card class or instance control scope

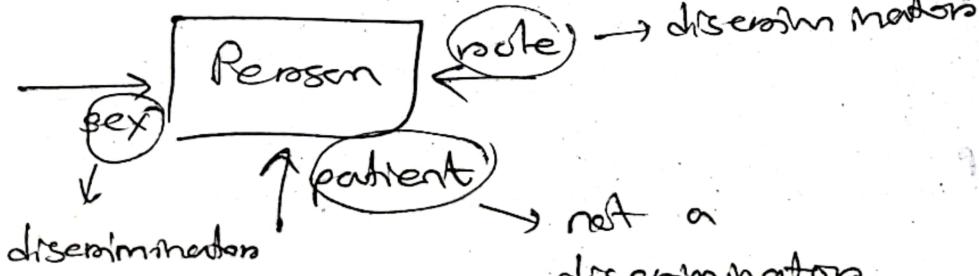
class scope → class to attribute manipulate

09/07/23

Multiple inheritance

Person ← Doctor ← Family Doctor

not a classification



dynamic classifications

- each status maintains its own set of contacts
- status go to depend on discriminators

changes often with respect to specific identifiers

these factors determine the priority (order)

Composite

- create, delete and move point (radius) delete

move

selection in algorithm

interface → skeleton / prototype

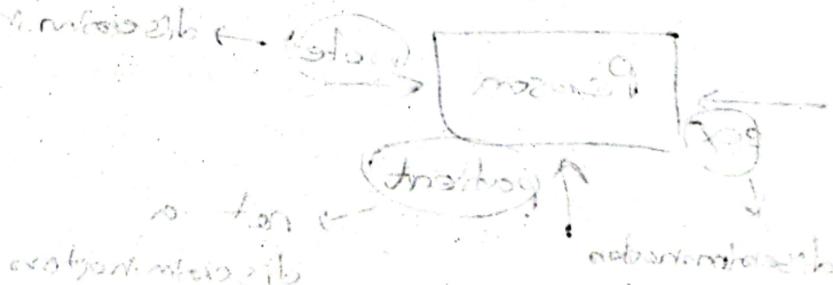
dimin → netball → basketball

basketball → basketball

---> : realization : Realisierung von

→ : generalization

♀ : lolipop



- when class uses some deep interface to one interface work, use the interface to other implementations more often
- consider dependency

09/07/
10/07/23

Boundary class

→ rep users input & deal

→ interfacing class

persistence

→ database engine / file

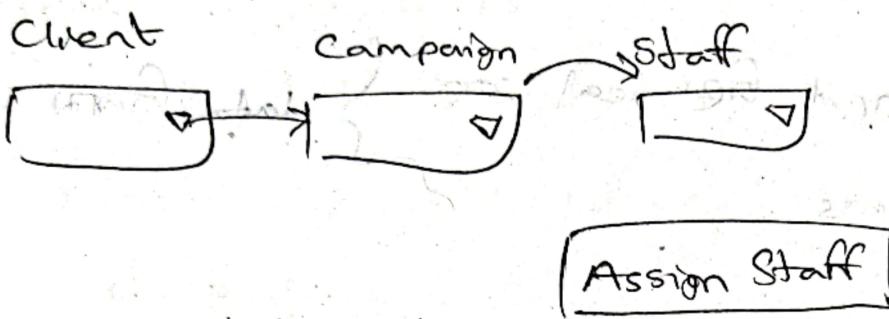
entity class → data

→ store data

→ db design

data processing → control class

15/07/23



client name → Client class

title → campaign title

staff class

message class

class diagram → accuracy of the system

→ just one propagation toward the system

→ after detailing more about design pattern will come

Finalized diagram ↴

9. getcampaign() X

getStaffCampaign ✓

class ↴
 |
 | properties
 |
 | operations

UML design

Use case design

Class diagram

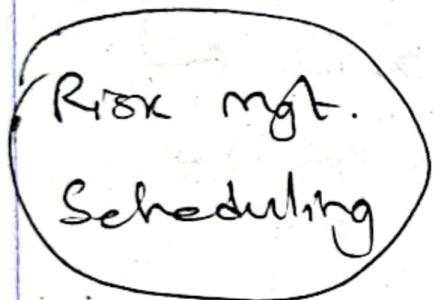
Advanced notation
of class diagram

Description → use case

↓
class diagram

05/08/23

100000



→ part of project Mgt.

PMI

Power PI

Business

Intelligence

Project managers → in charge of project mgt

Stakeholders → project & ~~in~~ related people

Closed paradigm → chain of command

Random paradigm → research organization

Work Breakdown Structure (WBS) ← decomposition

06/08/23

Software Project Estimation

man-month

man-day

man-hours

man-day → 8 hrs

man-month → 22 × 8 hrs

person

$$\text{man-day} = 8 \text{ man-hours}$$

$$\text{man-month} = 22 \text{ man-days}$$

person-hours

person-day → depends on complexity

Cost Estimation

cost estimation → requirement analysis

→ requirement analysis → change

just estimation → may not be accurate

effort → (EAW) Workload measurement

→ 52 technique → estimation range → good

business context

→ system → business logic

research → no boundary

project → boundary

↳ boundary set by us

size estimate

↳ Line of Code (Loc) → can measure it

↳ FP (functionality) → language to language
differ here

↳ function point / feature point

Functional decomposition

→ verb identify

→ find areas decompose, 20% 92/3 clear idea

Loc / pm →

→ person-month

→ person month per month 22x8 hrs

Person involved

Loc

Efforts (months)

7.4

24.4

30.9

13.9

Person involved

1

3

3

2

Month required

7.4

8 days each

10 days long

6

max - 12-20

final time

(considering
parallelism)

FP weights → company specific

complexity multipliers → project specific

0.25 p-m / FP

→ 0.25 function point implement ചെല്ലു തന്നെ മാറ്റുണ്ട്.

0.25 month work

exponent

curve-fitting are 4329

$$\text{effort} = \text{tuning coeff} \times t^{\text{exp}}$$

$$E = A (\text{Loc})$$

Bailey - Basili

→ 5.5 → min. effort one software is

cocomo model

→ constructive cost model

07/08/23

cocomo Model

Empirical model

System ৰাখলৈ no clear conception

→ Application Decomposition Model

Early Design Stage Model

→ অনেক ক্ষেত্ৰে better information আছেৰ বাবে এই মডেল

Reuse Model

→ ইন্টাৰ্ফেস implementation দি ফিল কৰিব

→ অবি পৰা অন্তৰ্ভুক্ত

→ অনুসূচি কি scratch দি কৰি আগো আজল, এমনি অনুসূচি কৰি code কৰি → reuse → কো অপি reuse,

→ যদি ফিল অচীক্ষা কৰি → cost কম

Post-architecture stage model

→ অনুসূচি কৰি অন্তৰ্ভুক্ত refined data এবং অনুসূচি কৰি কৰি কৰি

Early Design Model

Size = LoC

Reuse Model

Black box reuse → component ରୁ କିମ୍ବା ଏକ ଅଳ୍ପ କାମ କରି
 → ଫିଲ୍ ଚାର୍ଜ କରିବାର ପାଇଁ ଦେବାତା ନେଇ

White box Reuse → code ଓ change କରାଯାଇବାର ପାଇଁ

→ code କରାଯାଇବାର ପାଇଁ

Post

Sum
 Precededness → କାମ କରି କିମ୍ବା କରାଯାଇବାର କିମ୍ବା

Team cohesion → team ଏକାଥିର କାମ କରାଯାଇବାର କିମ୍ବା

process maturity → org. mature ହେଲୁଏ କିମ୍ବା

$$1.07 + \begin{pmatrix} 0.05 \\ 0.25 \end{pmatrix} \quad \left\{ 1.06 \text{ to } 1.26 \right.$$

Software eng'

→ system ରୁ ସମ୍ପର୍କ → P ରୁ ରୁକ୍ଷ

telecomm → reliability → ଆବଳା କାମ → ସମ୍ପର୍କ

embedded system → framework ରୁକ୍ଷ, assembly level

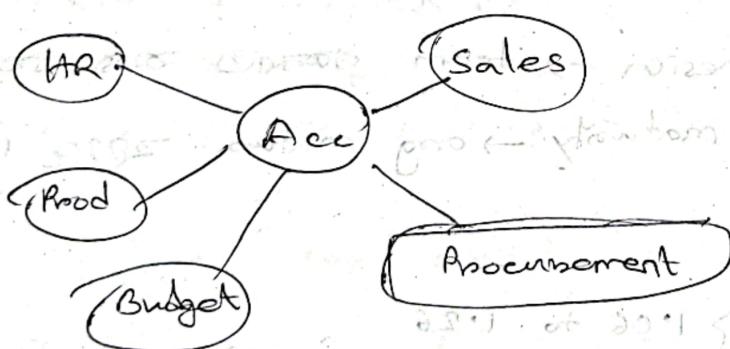
code ରୁକ୍ଷ ରୁକ୍ଷ ରୁକ୍ଷ

XX XX

Make-Buy Decision

COTS → reuse
↓
off the shelf

- ERP software → used generally & does s/w design w/o
- Enterprise Resource Planning software
 - accounting module
 - HR
 - production
 - budget
 - Sales
 - Procurement
- ERP configure
- demand expert
- Above programming?



SAP
↓
customized ERP w/o
Oracle
↓
ERP supplier

SAP →

- cost of 3 branches multiply
- sum of 3 branches = cost
- probability - 3 chance estimate of 20%

simple, difficult, 3n → 3D branch

probability sum = 1

Microsafe
Dynamics

12/08/23

Software Quality

SQA → Software Quality Assurance

Book

(South Indian written)

system follows documentation

→ manual

→ system & manual wise

→ Once software, then quality

excellent

requirement fulfill හැමත තුන් valid s/w - නේ මෙයි
quality - නෑ කාන් නෑ මෙයි නෑ

variation control → SQA නෑ

→ පරිග්‍රහ තේ, පරිග්‍රහ තේ පරිග්‍රහ තේ

→ පරිග්‍රහ තේ requirement නෑ මෙයි, quality නෑ

testing vs test case design

↓ difficult

easy

SQA is not testing

SQA → feedback and continual improvement

→ standard manual → standard practice follow රුජ්‍ය මූල්‍ය මිශ්‍ර

→ මෙයි requirement නෑ පරිග්‍රහ නෑ

improve තේ මාරු

quality test & flaw

- time रात्रि कान्धा (flaw detection efficiency)
- everyone is involved

flaw analysis

- काम पूर्ण हो तो एवं अन्तर्भूत विषय
- एक Step के बाद उसे पुनः करें और repeat

प्र 22

13/08/23

level 1 → essential requirements

→ उनीं चाहीं S/W करने वाले न

(जो आवश्यक गतिशील होंगे)

quality information

→ slide 19 first example correct ना

→ → ये failure मिस करे नहीं significant नियम ना,
असर नहीं रखे नहीं

→ "must repair" नहीं logically correct ना

cosmetic errors → ऐसे errors जो कैसी होनी चाहीं नहीं

→ e.g. font change, etc.

→ जो कैसी होनी चाहीं नहीं उनको नहीं बदलना चाहिए

→ नियम नहीं, फ्रैमप्रेस नियम नहीं, इत्यादि

Portability, reusability, interoperability \rightarrow quality factors

expresses the quality

transition \rightarrow implementation related

operation \hookrightarrow development related

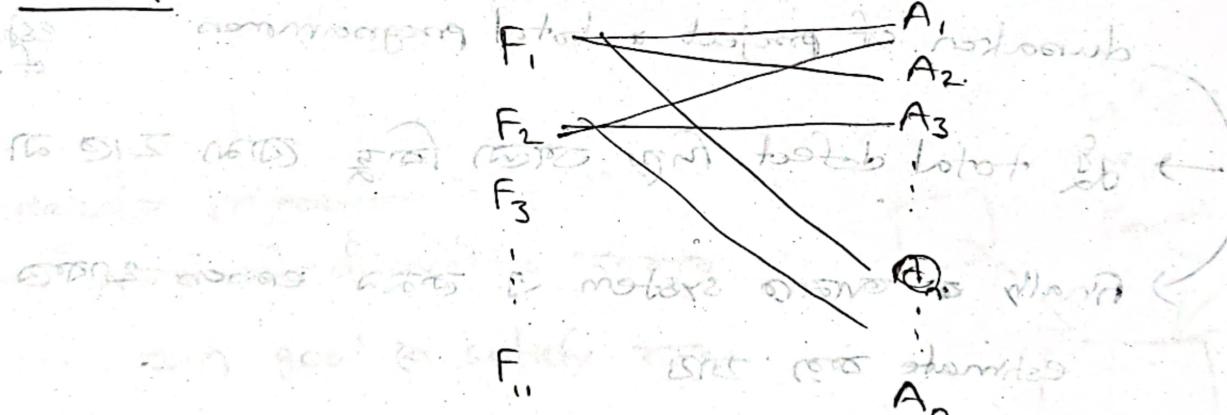
revision \rightarrow more change \Rightarrow poor partitioning

good quality address poor

Group

Factors

Attribute



McCall \rightarrow factors for quality

relationship attribute \hookrightarrow express

reusability \rightarrow modularity

McCall
1970

$$F_1 \Leftrightarrow A_1, A_2, A_3, A_4 \\ F_2 \Leftrightarrow A_2, A_3, A_5$$

$\left\{ \begin{array}{l} \text{overlapping} \\ \text{modular} \end{array} \right.$

metre → measurement of a system

defect → attribute

→ measure of efficiency of coding

→ total number of defects / duration

→ no. team members (no month worked) defect rate

→ to find out efficiency
of the programme

total defects

duration of project * total programmers

estimation
of future
projects

→ ~~total defect~~ ~~for~~ ~~then~~ ~~for~~ ~~then~~ ~~are~~ ~~in~~

→ Finally ~~on~~ system ~~in~~ ~~then~~ ~~errors~~ ~~have~~ ~~are~~
estimate ~~to~~ ~~be~~

→ better estimation

measure → quantification of attribute

→ ~~for~~ ~~one~~ ~~attribute~~ ~~or~~ ~~combination~~ ~~quantification~~

→ metric

→ ~~one~~ ~~/~~ ~~similar~~ ~~metric~~ ~~or~~ ~~quantification~~ ~~/~~ ~~unification~~ ~~→~~ ~~Indicators~~

data should be automatically collected

from some programs

effort →

Goal^b, definition template ~~Goal~~^b ~~uses~~ ~~in~~ resources

Analyze → efficiency of programmes

for the purpose of → correct estimation

w.r.t. → the coding ~~done~~ ^{done} for project

viewpoint → team lead

context → current project → automated trading system
(say)

financial standard

14/08/23

Goal^b ~~uses~~ ~~in~~ resources → ~~uses~~ ~~in~~ ~~for~~ ~~the~~ ~~purpose~~ ~~of~~ ~~the~~ ~~correct~~ ~~estimation~~
metrification program

→ ~~uses~~ ~~in~~ ~~for~~ ~~the~~ ~~purpose~~ ~~of~~ ~~the~~ ~~correct~~ ~~estimation~~ ~~metrification~~ ~~program~~

where → the goal to satisfy ~~the~~ ~~use~~ ~~in~~ ~~for~~ ~~the~~ ~~correct~~ ~~estimation~~

fan-in → ~~uses~~ ~~in~~ ~~for~~ ~~the~~ ~~purpose~~ ~~of~~ ~~the~~ ~~correct~~ ~~estimation~~ ~~metrification~~ ~~program~~

fan-out → ~~uses~~ ~~in~~ ~~for~~ ~~the~~ ~~purpose~~ ~~of~~ ~~the~~ ~~correct~~ ~~estimation~~ ~~metrification~~ ~~program~~

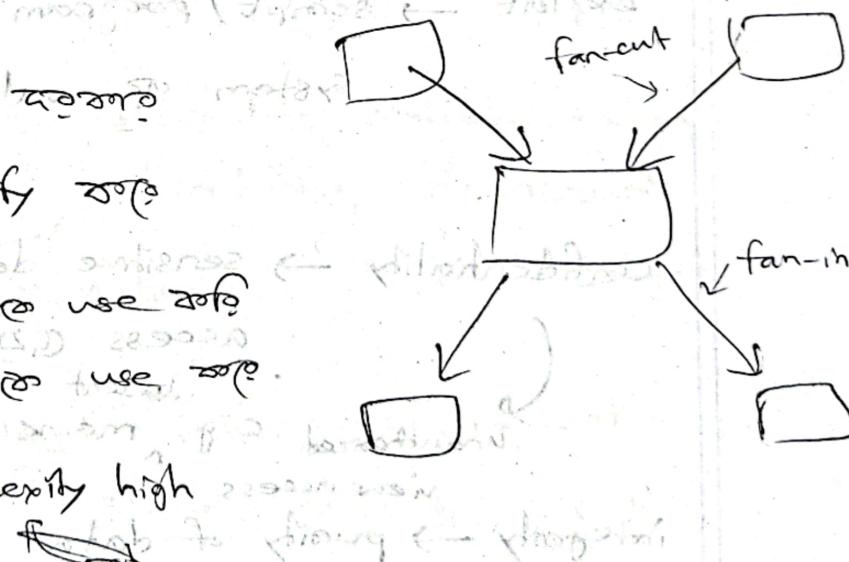
fan-out & fan-in ⇒ complexity high
~~fan-out & fan-in ⇒ complexity high~~

component and connection ⇒ architектurally complex

old-style, bottom-up

new-style, top-down

new-style, top-down, ~~old-style, bottom-up~~



cohesion → පාලන උස්සි අනුකූල මෙතහේද සංවර්ධන ප්‍රාග්‍රැම්ප්‍රාග්‍රැම් ප්‍රාග්‍රැම්

වෘත්තීය

→ multiple purpose හෝ ප්‍රාග්‍රැම් සේවක මෙතහේද ප්‍රාග්‍රැම් ප්‍රාග්‍රැම් ප්‍රාග්‍රැම්

වෘත්තීය

atomic operation → පාලන ආකෘති හෝ පාලන ආකෘති

වෘත්තීය

26/08/23

Quality framework

→ organization of quality support (प्रयोगशाला की सहायता)

CMMI → Capability Maturity Model Integration

QAI, ^{Institute} Quality Assurance Institute

→ ultimate purpose → overall improvement of business results

CMMI certification

→ assessing four levels

System → growth maturity of human resource

Initial → heroic approach

→ process follow

→ firefighting

→ reactive → इसका दर्जा नहीं अभियान के रूप में

Managed → proactive → इसका दर्जा नहीं अभियान के रूप में

→ इसका दर्जा नहीं अभियान के रूप में

→ defining project level → इसका दर्जा नहीं अभियान के रूप में

manager के रूप में

Planning

process

project management, process

management

Defined → organisational level → process follow

→ procedure → ~~order for action~~

→ इसका दर्जा नहीं अभियान के रूप में

→ training के रूप में done → productivity बढ़ाव

tool management (buying, installation) etc. → cost related

→ level 3 → throughout the organization ഒരു രീതി

mark down ചെയ്യാൻ തന്നെ കൗൺസിൽ വിലക്ക് എന്നും

→ ഒരു രീതി എന്നതുമേൽ കൂടി മറ്റ്, അതാണ തന്നെ

പോലെ എ

→ extended to fragmentation, teams etc., assembly line style etc.

level 4 → measurement, control

level 5 → change your process / improve

→ optimal എന്നും level 1 beach എന്നും

level 6 → level $(i-1)$ ഓഫ് ടൈം അപ്പോൾ, അല്ലെങ്കിൽ

അതിനു തുടർച്ചയാണ് എന്നും അതിനു പുനരുപയോഗിക്കാൻ കൂടിയാണ് എന്നും അതിനു പുനരുപയോഗിക്കാൻ കൂടിയാണ്

level 3 → 11 processes

→ 6 processes → $\textcircled{7+5}$ \rightarrow 12 main driving forces

resulting in a local leadership model at a higher level

27/08/23

Source Control System

- ↳ source code
- ↳ shared by multiple programmers
- source code version द्वारा यहाँ अपने एक संस्करण होता है
- अलग programmers ने भी काम करते हैं
- project का multiple version थार्ड नहीं होता है

MS TFS
↓
Azure

Version control

- shared source code and files
- (VCS) Version control system
- change management
- कोई किसी change को बदलने के लिए → change request
- request approved होने के बाद change को बदला जाता है

git
SVN
TFS
JIRA

revision → state of a document at a particular point in time

- checkout / clone → repository के file local machine पर आते हैं
- distributed machine वे lock नहीं होते तो machine connection lost हो जाएँ तो file always lock होते हैं

Lock - Modify - Unlock Model

→ ~~can~~ merge conflict ~~in~~

Copy - Modify - Merge Model → check out \rightarrow 1 file

→ locally merge ~~can~~ merge conflict ~~in~~ ~~with~~ ~~with~~

→ ~~can~~ commit conflict ~~in~~ ~~with~~

~~→ If~~ ~~or~~ ~~third~~ ~~of~~ ~~operations~~ ~~meanwhile~~ Use Case
VR design

→ say B commit ~~in~~

→ A commit ~~in~~ ~~in~~ ~~in~~

→ locally merge ~~in~~ (A+B)

→ meanwhile C 3 locally merge ~~in~~ (B+C) and commit

→ (A+B) commit ~~in~~ ~~in~~ conflict ~~in~~

→ (B+C) ~~in~~ ~~in~~ conflict ~~in~~ ~~in~~ ~~in~~

DVE

→ clone \rightarrow multiple files

→ ~~use~~ conflict, ~~use~~ fetch [20 min]

→ given a scenario

→ of the ~~in~~ operators

~~use~~ describe

→ ~~use~~ codebase locally save ~~in~~

→ compile and run ~~in~~ ~~in~~ helpful

[slide]

Beta 2 release ~~in~~ ~~in~~

→ file A ~~in~~ change ~~in~~ 1.4 ~~in~~ ~~in~~

→ file B ~~in~~ ~~in~~ 1.2 ~~in~~ ~~in~~

→ file C ~~in~~ change ~~in~~ 2.1

{ tag \rightarrow file ~~in~~ ~~in~~ situation ∞ covers ~~in~~