Syllabus

1. Software Engineering

2. Software Life cycle model

3. Software planning

4. Requirement : Analysis and Specification

5. Software Design and Implementation

Uttar Pradesh Polytechnic (YouTube)

1

Software Engineering

Software Engineering ⇒ Software-

Engineering का अर्थ एक ऐसी Engineering से है, जिसमें Computer System तथा किसी अन्य इलेक्ट्रॉनिक Device के लिए Software का निर्माण किया जाता है,

" Software Engineering एक प्रक्रिया है, जिसके user की जरूरतों को Analyze किया जाता है, और उन जरूरतों को Analyze करके उसके आधार पर Software को बनाया जाता है,

Uttar Pradesh Polytechnic (YouTube)

* सामान्यतः किसी इलेक्ट्रॉनिक उपकरण या गैजेट में दो प्रकार के अवयव होते हैं,

1. Hardware

2. Software

1- Hardware ⇒

Hardware एक भौतिक तत्व है, जिसे देखा और छुआ जा सकता है, जो Computer system को चालू करने के लिए संयोजक का कार्य करते है,

Example- computer का मॉनीटर, keybord. mouse, Hard Disk etc.

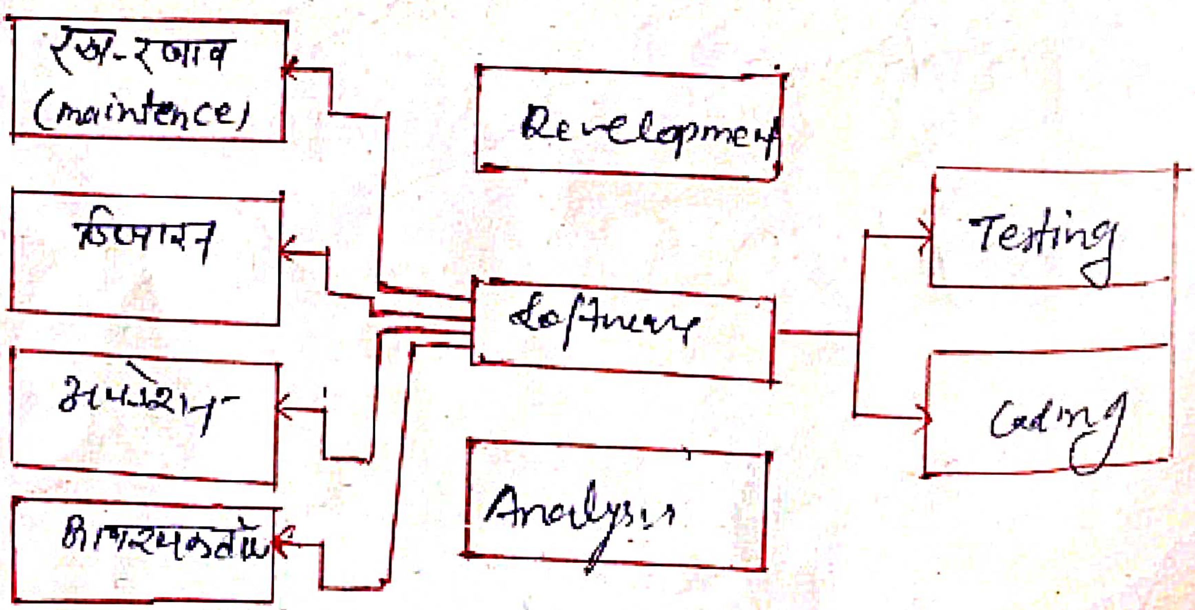
Software ⇒

Software, निर्देशों, Data और programo का एक संग्रह है, जो किसी विशिष्ट कार्य का करने के लिए बनाया जाता है,

Ex- MS office, window media player etc

Uttar Pradesh Polytechnic (YouTube)

Software Engineering of Definition



Software Engineering का उद्देश्य

1. रख-रखाव (maintainability)
2. जरूरतों को सही तरीके से लागू करना
3. पुन प्रयोग (Reusability)
4. Testablity (परीक्षण योग्य)
5. विश्वसनीयता (Reliablity)
6. portebillity
7. Adaptebility (अनुकूलन क्षमता)

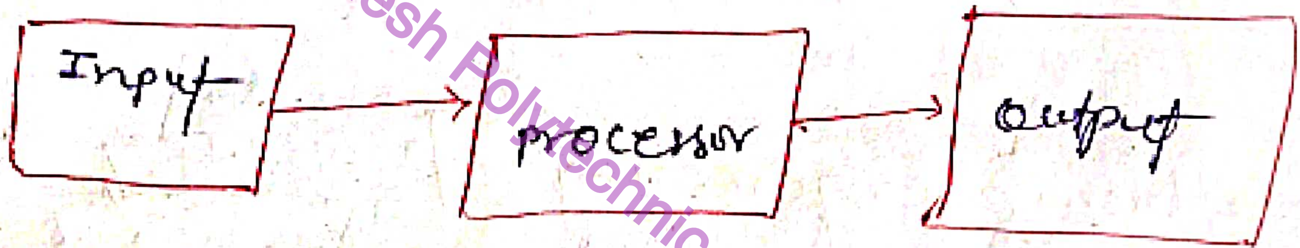
Uttar Pradesh Polytechnic (YouTube)

Software Engineering की जरूरत ?

1. बडे Software का प्रबंध
2. लागत प्रभावशीलता
3. प्रकृति का प्रबंध करना
4. परीक्षण की प्रक्रिया आसान बनाना

(4) Software System

Software system एक सम्पूर्ण system है, जो सूचनाओं के आदान-प्रदान और उन्हे साझा करने के काम आता है,
साधारणतया किसी कम्प्यूटर system मे हम processor को एक INPUT देते है, जिसे वह output मे बदल देता है,

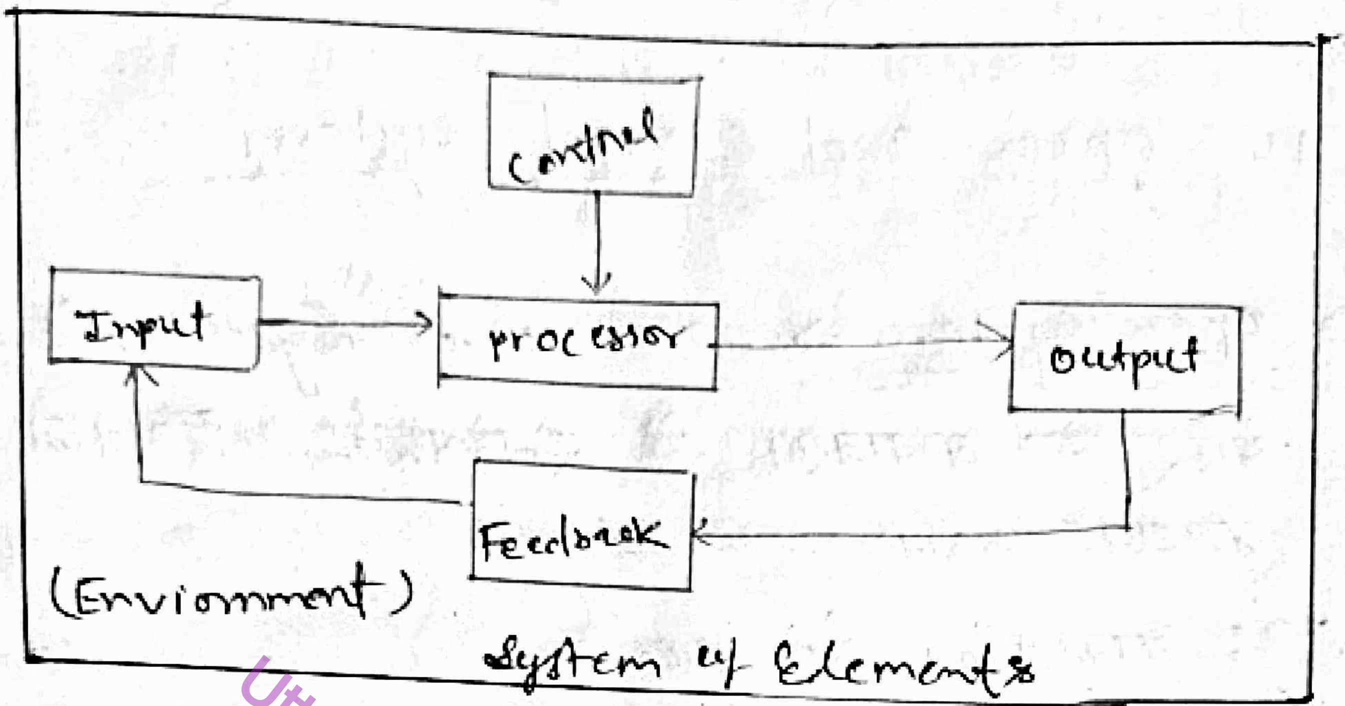


Feature of Software System

(1) Organized (संगठित)
(2) Interaction (सम्बंधता)
(3) Integration (एकीकरण)
(4) Inter dependent (परस्पर निर्भरता)

Uttar Pradesh Polytechnic (YouTube)

Software System of Elements D/4



1. Input → Input यह Data या सूचनाएँ है, जिन्हें System मे भेजा जाता है, जिसे processed करने जरूरत के लिए उपयोगी सूचना मे बदला जाता है,

2. processor → सूचनाओं को इकट्ठा करता है, और उसे काम करने सूचनाओं को जरूरत जैसे बदलना

3. Output → यह processor से प्राप्त परिणाम होता है,

4. नियंत्रण → एक प्रोसेसर के संलग्न को operating system और software द्वारा नियंत्रित किया जा सकता है,

5. Feedback → यह एक Dynamic System है, जिसके लिए नियंत्रण का उपयोग होता है.

6. Environment → यह system के कार्य करने का वातावरण है/

Uttar Pradesh Polytechnic (YouTube)

System के Types

1. open and closed system

Open System ⇒ यह ऐसा System है जो बाहर के वातावरण से स्वतंत्रतापूर्वक सूचनाओं का आदान प्रदान करता है,
अतः वातावरण में बदलाव होने पर System में बदलाव करना पड़ता है।

Closed System ⇒ एक closed System Environment के साथ Interaction का आदान-प्रदान नहीं करता,
वातावरण में होने वाले बदलाव इसे प्रभावित नहीं करते हैं,
इस तरह के system बहुत कम होते हैं।

(2) Static (स्थिर) और गतिशील (Dynamic System)

Static स्थिर System ⇒ यह ऐसे system होते हैं जहाँ, वर्तमान value पर निर्भर करता है
यह भौतिक रूप से विश्वसनीय होते हैं।

Uttar Pradesh Polytechnic (YouTube)

Different between program & Software product:

program	Software product
1- यह आकार मे छोटे होते है	1- आकार मे बड़े होते है,
2- यह एक Developer द्वारा विकसित किया जा सकता है,	2- इसके लिए developers की टीम की आवश्यकता होती है
3- इनका उपयोगकर्ता (user) से Interface सही ढंग से नही है,	3- इनके उचित उपयोगकर्ता Interface की सुविधा होती है,
4- एक program software नहीं हो सकता,	4- एक software program हो सकता है,
5- size किलोबाइट (kb) से लेकर मेगाबाइट (mb) तक हो सकता है	5- इसका size मेगाबाइट से लेकर गीगाबाइट (Gb) तक हो सकता है
6- program की कार्यक्षमता compiler पर निर्भर होती है।	6- इसकी कार्यक्षमता OS पर निर्भर करती है,
7- program की कोड लाइनों (code line) के संख्या कम होती है।	7- software मे कोडलाइन बहुत अधिक होती है,

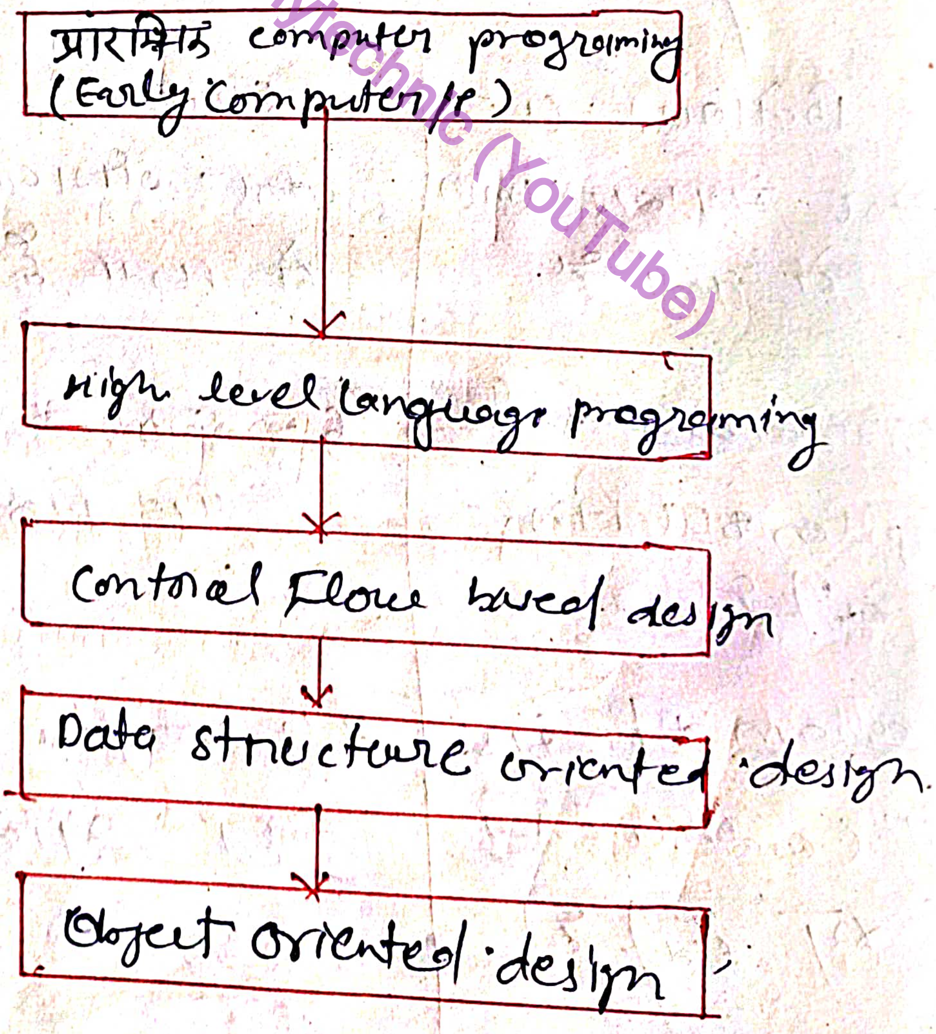
Uttar Pradesh Polytechnic (YouTube)

(2) Emergence of software engineering

(सॉफ्टवेयर इंजीनियरिंग का उद्भव)

Software Engg. की शुरुआत प्रारम्भिक Computer programing से होकर उच्च स्तरीय programing भाषा; उसके बाद नियंत्रण प्रवाह आधारित डिजाइन (Control flow design) से होकर Data संरचना पर आधारित डिजाइन और फिर object oriented design तकनीक आती है, जो Software विकास की प्रक्रिया में उभरती है।

Uttar Pradesh Polytechnic (YouTube)



1. प्रारम्भिक कम्प्यूटर प्रोग्रामिंग (Early Computer programing) ⇒

प्रारम्भिक computer programming में Assembly Language का इस्तेमाल किया जाता था, जिसे Low Level Language कहा जाता है, यह मशीन code पर आधारित था, उस समय गति और मेमोरी क्षमता के कारण programmers केवल Assembly भाषाओं को लिखने के लिए मजबूर थे, जब कभी का जल्दी ही एहसास हुआ कि low level Language के लिए बहुत मेहनत और बौद्धिक क्षमता की आवश्यकता थी, जिसके कारण High level Language का निर्माण हुआ,

Uttar Pradesh Polytechnic (YouTube)

2-> High Level Language (उच्च स्तरीय प्रोग्रामिंग भाषा)

पहले low level programming language की जगह लेने लगी, ज्यादा गति में प्रगति होने लगी, यह Compiler का उपयोग करती थी, फोरट्रान और कोबोल जैसी प्रारम्भिक उच्च स्तरीय भाषाओं के निर्माण के बाद कई भाषा C, C++, Pascal, Python, & Java etc. अस्तित्व में आयी,

Feature of High level programming Language

- मानव भाषाओं के समान समझने मे आसान है,
- Program के अनुकूल होती है, और Code करने में आसान होती है,
- समझने में आसान है,
- संकलन (Compile) & Debug में आसान होती है

Translator

1. Interpreter — कोड → इंटरप्रेटर → O/P
2. Complier — Code → Complier → Machine Code → O/P
3. Assembler

(3) Control Flow Based Design

(नियंत्रण प्रवाह आधारित डिजाइन)

High level Language के आने के बाद Computer का उपयोग तेजी से बढ़ने लगा,
Program ज्यादा जटिल और बड़े होने लगे
जिससे उनका प्रबंधन मुश्किल होने लगा।
इसे आसान बनाने के लिए Control Flow Based Design तैयार किया गया;

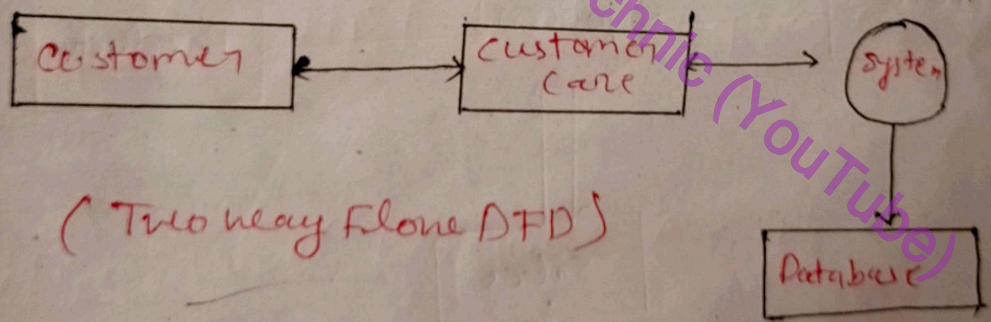
Uttar Pradesh Polytechnic (YouTube)

④ Data Flow Oriented design

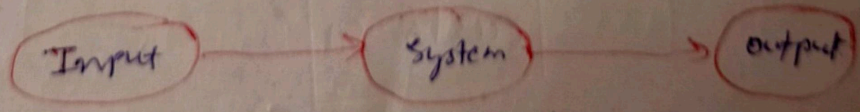
Networking और ग्राफिकल यूजर इंटरफेस (GUI) जैसी तकनीको के विकास के साथ जटिल Software को Control Flow आधारित डिजाइन से नियंत्रित करना मुश्किल हो गया.

अतः एक नई तकनीक Data flow Design अस्तित्व मे लाया गया,

"Data Flow design एक ऐसी तकनीक है, जिसका उपयोग, Computer Application program के अलग-2 हिस्सो से निश्चित जानकारी को एकत्र करने के लिये किया जाता है



(Two way Flow DFD)

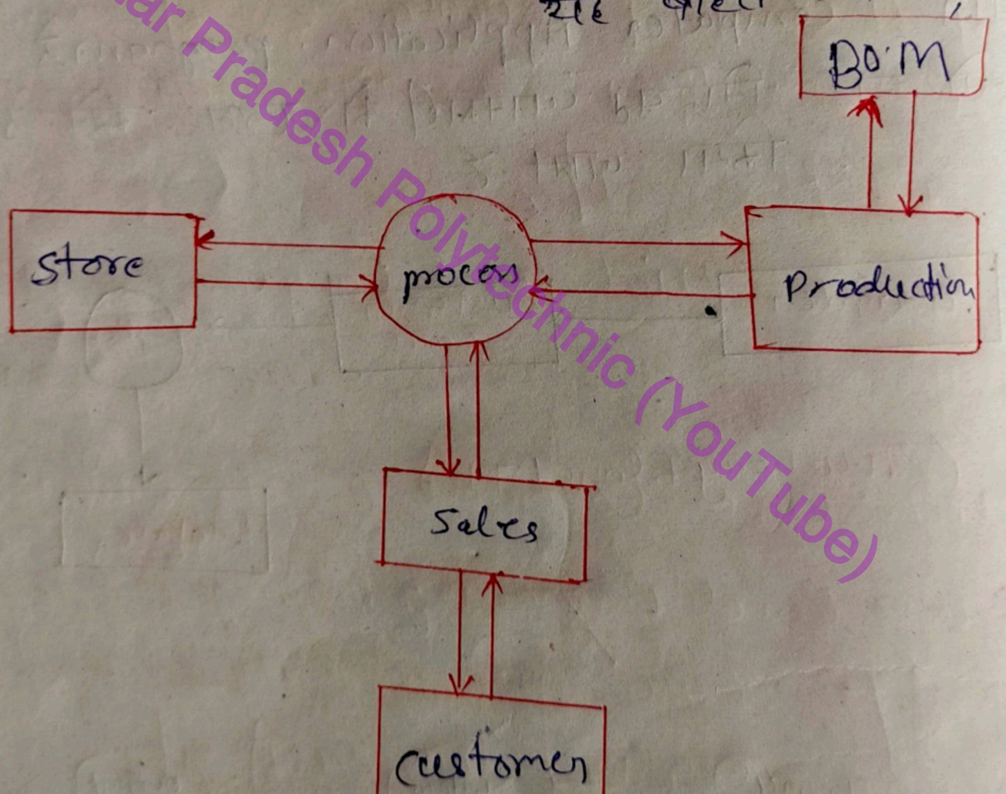


one way Data flow

Uttar Pradesh Polytechnic (YouTube)

Data Flow Diagram in some Elements

1. Entity → □
2. Process → ○
3. Data का प्रवाह → ⟶
4. Data base → ═
5. टर्मिनेटर (Terminator) → यह system का हिस्सा नहीं है, यह बाहरी तत्व है,



Data Flow D/G

Uttar Pradesh Polytechnic (YouTube)

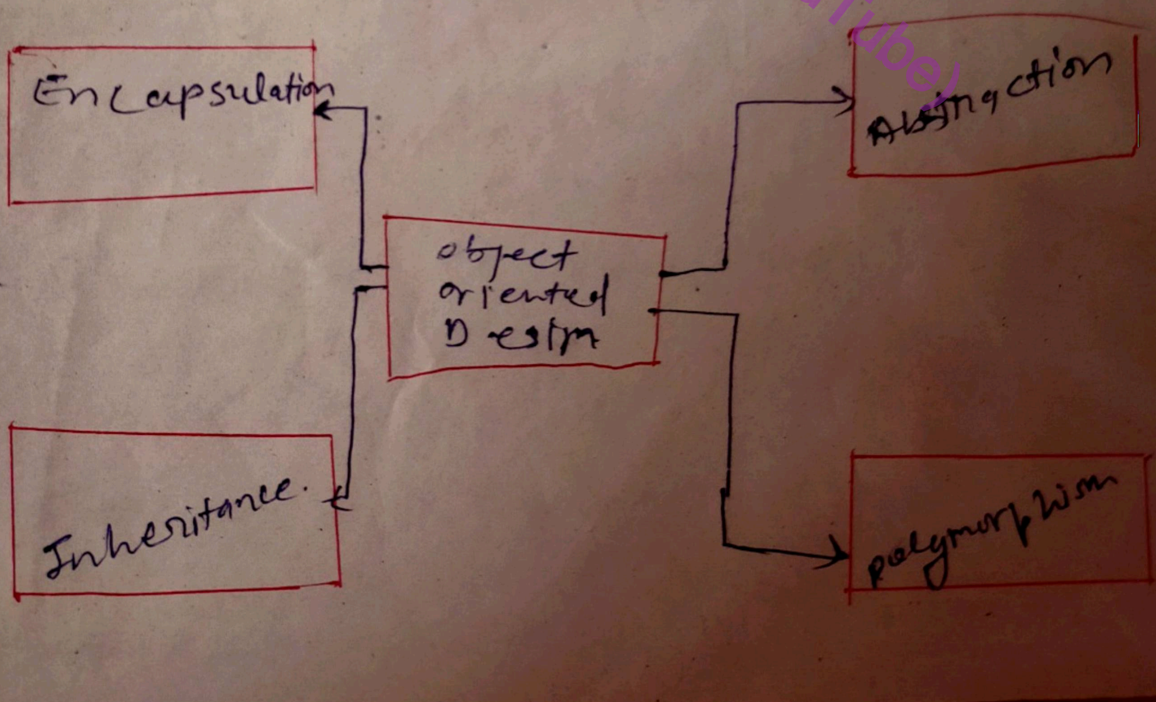
(5) Object oriented Design.

इसमें programic की संरचित तकनीक की विशेषताएँ भी है, और (Encapsulation), (abstraction), Inheritance, & polymorphism जैसी विशेषता है,

OOD में प्रयुक्त की जाने वाली शब्दावली

1) Object
2) Class
3) message
4) abstraction
5) Encapsulation
6) Inheritance
7) polymorphism

Uttar Pradesh Polytechnic (YouTube)



Software Life Cycle models

Chapter 2

जीवन चक्र मॉडल की आवश्यकताएँ ⇒

1) प्रक्रिया के पदानुक्रम के बारे मे डेवलपर्स के लिए स्पष्ट दिशा निर्देशो को परिभाषित करना।

2) गतिविधियो और डिलिवरेबल्स के एक मानक set के लिए रूपरेखा प्रदान करना

3) एक software program विकसित करने मे समय और लागत को कम करने के लिए।

4) हर कोई एक ही सिस्टम के तहत काम करे,

Software के विकास मे कई चरण शामिल है

1) Feasibility of software (सॉफ्टवेयर की व्यवहार्यता)

2) (Requirement gathering & analysis)

3) डिजाइन करना (Design Phase)

4) क्रियान्वयन करना (Implementation) या Coding

5) परिक्षण (Testing)

6) तैनाती (Deployement)

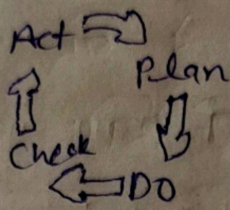
7) रखरखाव (maintenance)

Uttar Pradesh Polytechnic (YouTube)

* एक Software model design

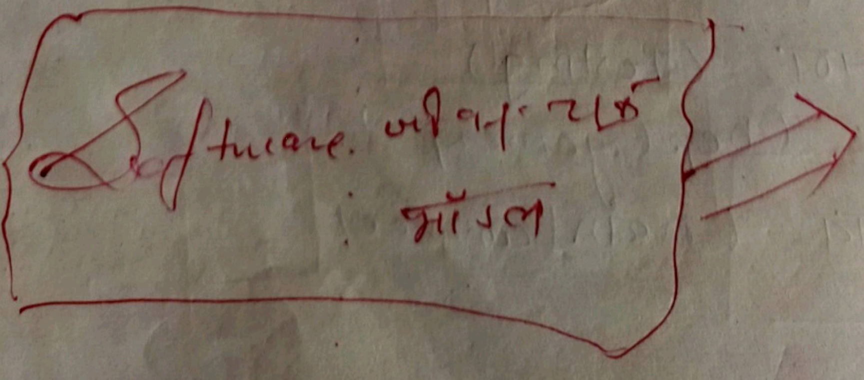
यह कुछ हद तक PDCA चक्र के समान है,

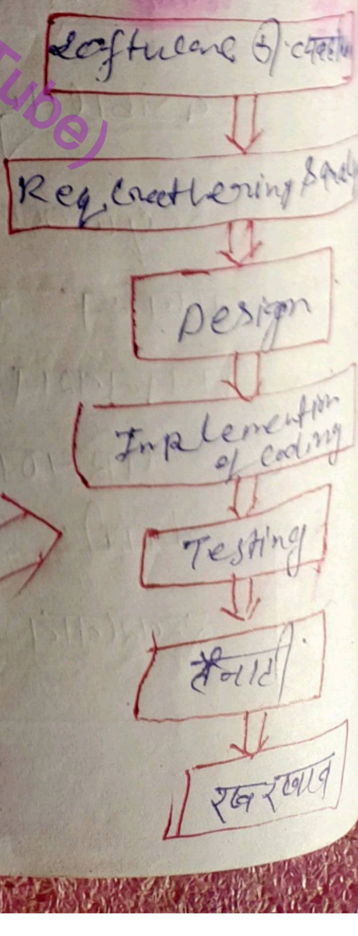
Plan, DO, Check, Act



* यह एक software विकास प्रक्रिया का एक सरल जीवन चक्र मॉडल है,

- Software की व्यवहार्यता (Feasibility of software)
- Req Gathering & analysis (आवश्यकता को एकठा करना)
- Design (डिजाइन)
- Implementation of coding (कोडिंग को लागू करना)
- Testing (परीक्षण)
- Deployment (तैनाती)
- maintainance (रखरखाव)





Uttar Pradesh Polytechnic (YouTube)

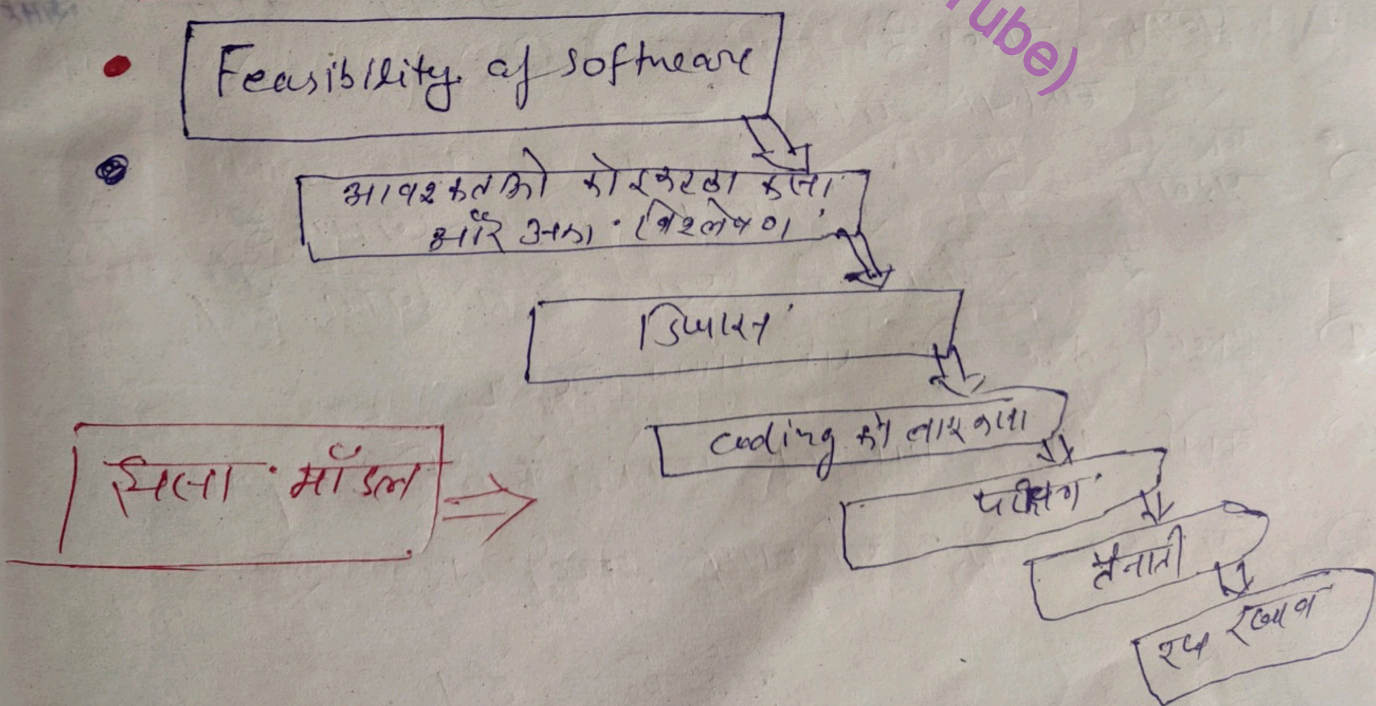
Software जीवन चक्र मॉडल के प्रकार

1. Classical waterfall model
2. Prototyping model
3. Evolutionary model
4. Spiral model

1. Classical waterfall model (वॉटरफॉल model)

झरना मॉडल रैखिक अनुक्रमित चरणों में विभाजित एक बुनियादी software model है, जहाँ प्रत्येक चरण पिछले चरण के विवरण पर निर्भर करता है, और कार्यों की विशेषताएँ से मेल जाता है, काफी हद तक एक दिशा मे बहता है एक दिशा मे बहने के कारण यह एक सीढ़ी की तरह दिखता है,

झरना शास्त्रीय मॉडल



Uttar Pradesh Polytechnic (YouTube)

झरना model का अनुप्रयोग

- हर विकसित software अलग है, और इसके लिए उपयुक्त (SDLC) दृष्टिकोण की आवश्यकता होती है जो आंतरिक और बाहरी कारकों पर निर्भर करता है।

- उत्पाद की परिभाषा स्थिर है,
- कोई अस्पष्ट आवश्यकताएँ नहीं है,
- प्रोजेक्ट छोटा है।
- उत्पाद के समर्थन के लिए आवश्यक विशेषता वाले पर्याप्त संसाधन उपलब्ध है।

झरना model Advantage

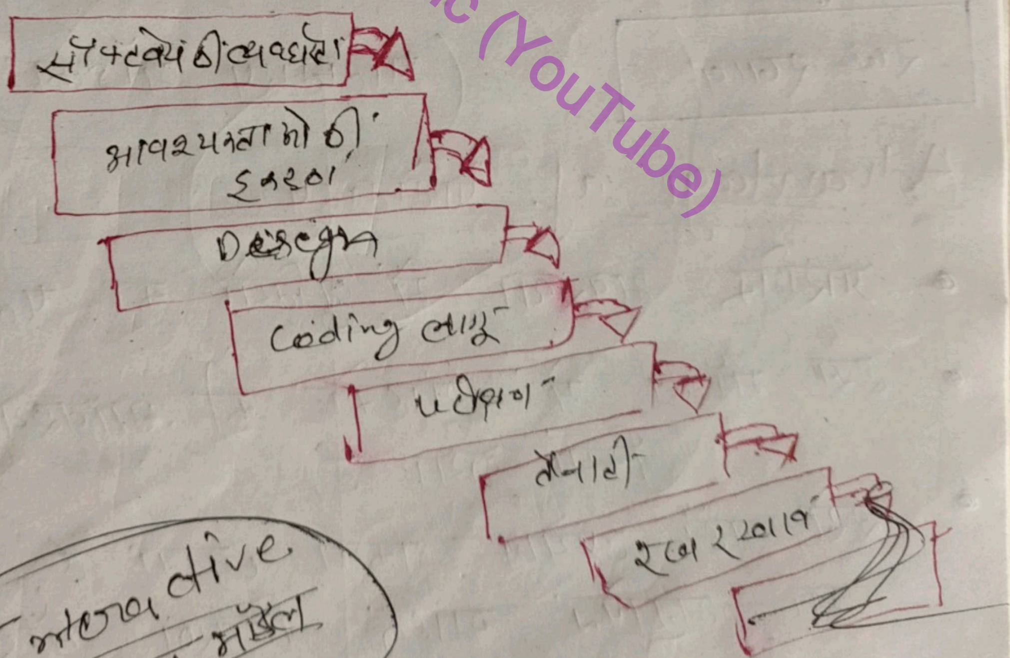
1. सरल और समझने और उपयोग में आसान
2. मॉडल की दृढ़ता के कारण प्रबंधन करना आसान प्रत्येक चरण के विशिष्ट डिलिवरेबल्स और एक समीक्षा प्रक्रिया है
3. चरणों को संसाधित किया जाता है और एक समय में पूरा किया जाता है।
4. अच्छी तरह से परिभाषित उद्देश्य
5. कार्यों को व्यवस्थित करने के लिए आसान
6. प्रक्रिया और परिणाम अच्छी तरह से प्रलेखित है।

Uttar Pradesh Polytechnic (YouTube)

Iterative झरना model: Advantage

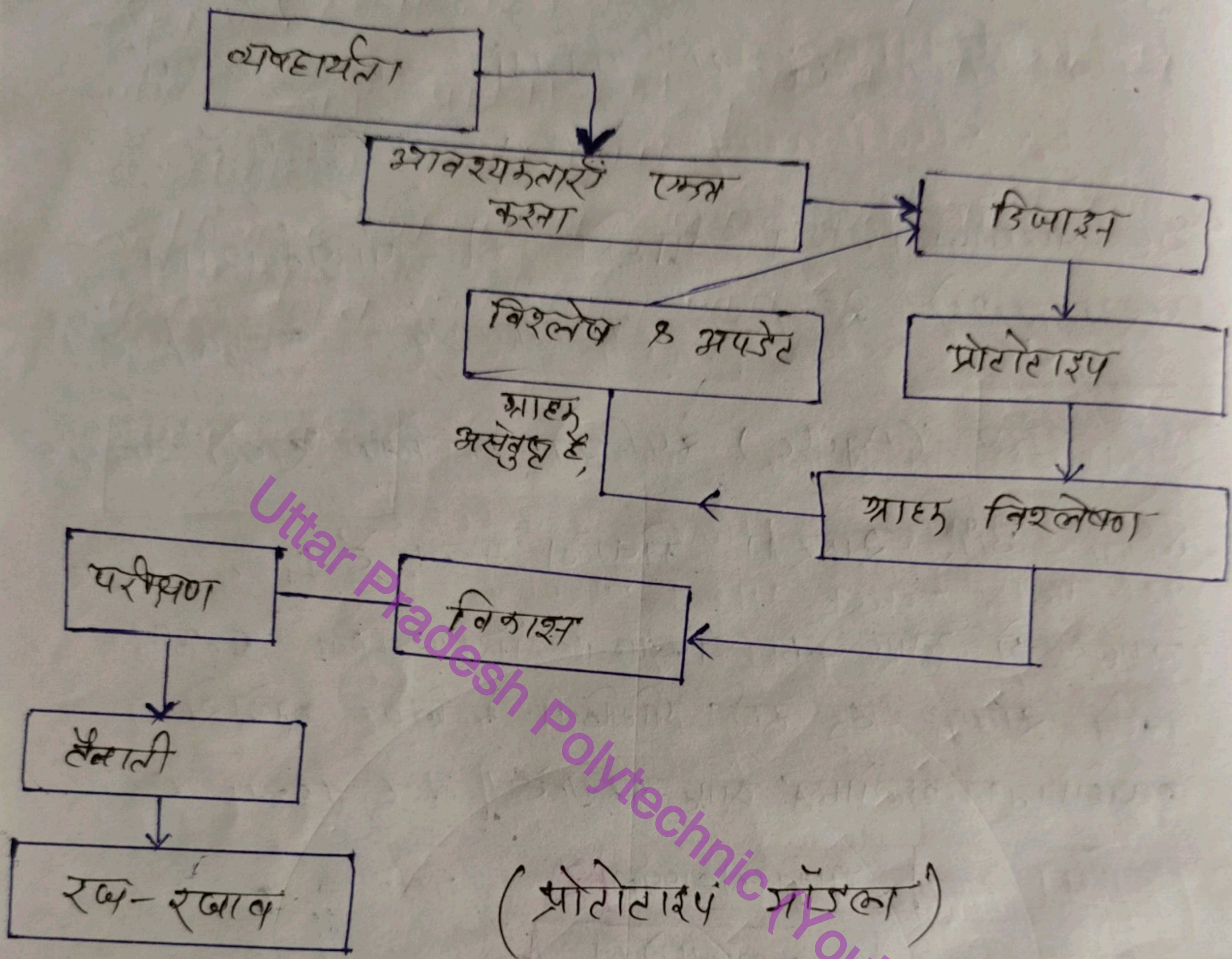
- कुछ रोजगार योग्य क्षमताओं को Software विकास जीवन चक्र में अग्रिम रूप से बनाया जा सकता है,
- यह परियोजना या ग्राहकों की परिवर्तनशील आवश्यकताओं के लिए कुशलतापूर्वक बहुमुखी है।
- यह चुस्त (Agile) कंपनियों के लिए सबसे उपयुक्त है,
- जाँच और समस्या निवारण सरल है।
- ग्राहक को उत्पाद का डिजाइन और रूपरेखा प्रदर्शित करते समय हर कोई प्रक्रिया के लिए भरोसेमंद उपभोक्ता मूल्यांकन प्राप्त करता है।

Uttar Pradesh Polytechnic (YouTube)



Iterative झरना मॉडल

② प्रोटोटाइपिंग मॉडल



(प्रोटोटाइप मॉडल)

Uttar Pradesh Polytechnic (YouTube)

Advantage of प्रोटोटाइप

- प्रारंभिक अवस्था में त्रुटियों का पता लगाया जाता है।
- इस मॉडल में ग्राहकों की आवश्यकता को मान्य किया गया है,
- इसे वहाँ उपयोग किया जाता है जहाँ अपूर्ण या अनुचित ग्राहक विवरण मौजूद है।
- मान्य विनिर्देशों को आवश्यकता विश्लेषण

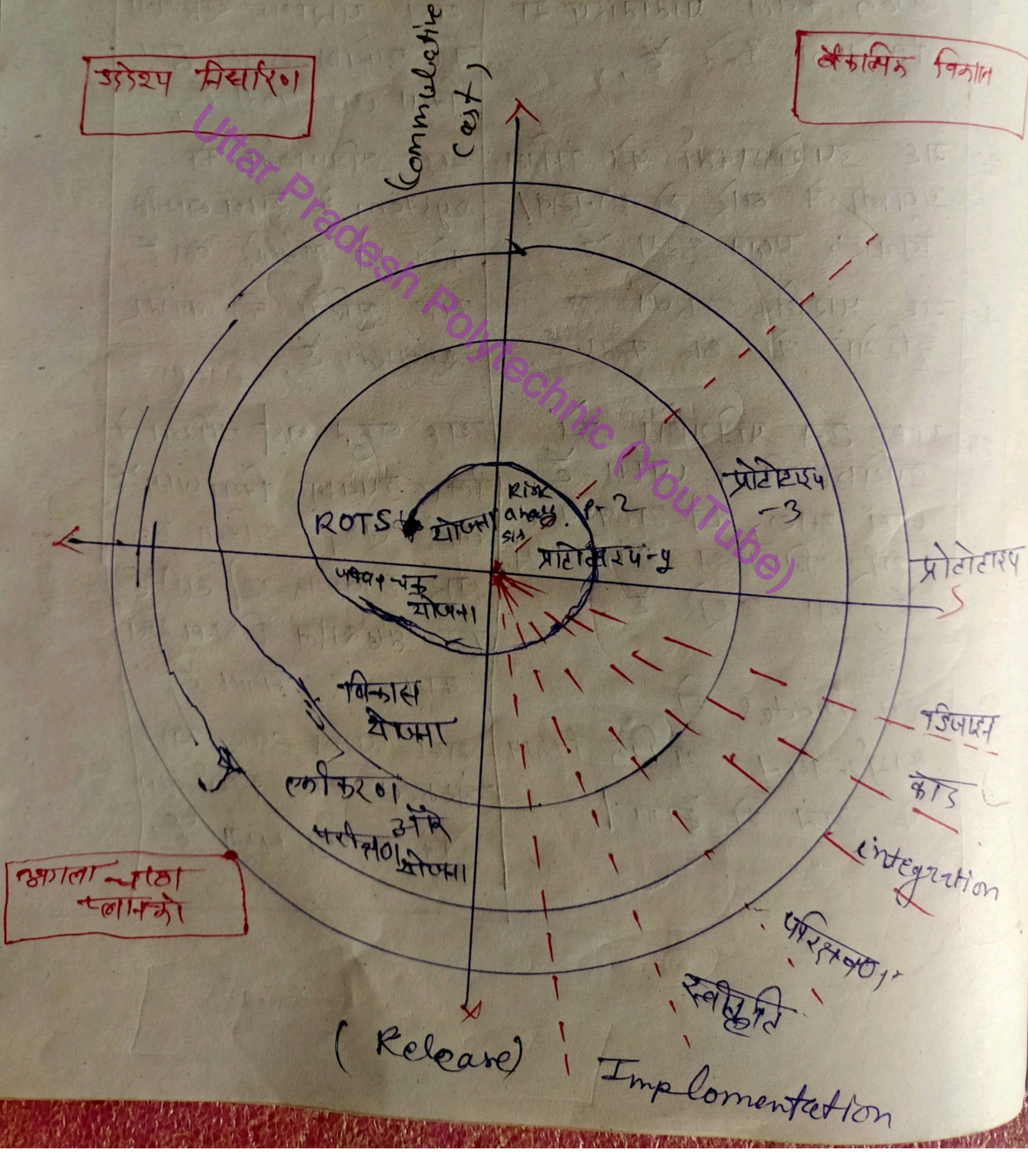
Different between – प्रोटोटाइप मॉडल and विकासवादी मॉडल

N	प्रोटोटाइप मॉडल	विकासवादी मॉडल
1-	यह एक पुनरावृत्त प्रकार का मॉडल है, डिजाइन ग्राहक की प्रतिक्रिया के विश्लेषण के आधार पर उत्तरोत्तर बेहतर होते जाते हैं	यह वृद्धिशील प्रकार है, डिजाइन विभिन्न डिजाइनो का विकसित रूप या संयोजन है।
2-	ग्राहक केवल प्रोटोटाइप का विश्लेषण करते हैं,	यहाँ प्रारंभिक संस्करणो का उपयोग ग्राहक द्वारा किया जा सकता है,
3>	यह उपयोगकर्ता को अंतिम प्रणाली के बारे में अनुभव/विचार प्रदान करता है,	यह उपयोगकर्ता को system के साथ बातचीत करने की अनुमति देता है,
4>	यह प्रारंभिक चरणों में त्रुटियों को ठीक करता है,	यह त्रुटियों को लगातार सुधारता है,
5>	यह उन प्रक्रियाओं में उपयोग किया जाता है, जहाँ ग्राहक की आवश्यकता और विनिर्देश अंतिम नहीं है, या अधूरे हैं,	यह बहुत बड़ी प्रक्रिया के लिए उपयोग किया जाता है, जहाँ सभी प्रक्रिया एक बार आवश्यक नहीं होती है एक वृद्धिशील विकास यहाँ काम कर सकता है,
6>	इस model में कार्यान्वयन अंतिम विकास के अंत में होगा।	यहाँ चरण-वार या संस्करण-वार कार्यान्वयन किया जा सकता है

Uttar Pradesh Polytechnic (YouTube)

SPIRAL model

Spiral (घुमावदार) मॉडल में Software के जीवन चक्र प्रक्रिया का सर्पिल के रूप में प्रतिनिधित्व किया जाता है, Software प्रक्रिया के प्रत्येक चरण को सर्पिल के प्रत्येक घुमाव द्वारा दर्शाया जाता है,



Uttar Pradesh Polytechnic (YouTube)

उपयोग Spiral model कहाँ करे

- जब परियोजना बड़ी है,
- जब विभिन्न संस्करणों को जारी करना आवश्यक हो
- जब प्रोटोटाइप की आवश्यकता होती है,
- जब जोखिम मूल्यांकन महत्वपूर्ण है,
- जब प्रणाली को जटिल के रूप में डिजाइन किया जाना है,
- जब आवश्यकताओं में लचीलापन आवश्यक है,

Advantage of spiral model

1. सतत विकास जोखिम प्रबंधन मे मदद करता है,
2. विकास तेज है
3. सुविधाओं को एक व्यवस्थित तरीके से जोड़ा जाता है,
4. सभी Software विकास जीवन चक्र मॉडलो की सर्वोत्तम विशेषताओं को शामिल करता है,
5. प्रलेखन नियंत्रण मजबूत है,
6. परियोजना का अनुमान अधिक सटीक होगा,

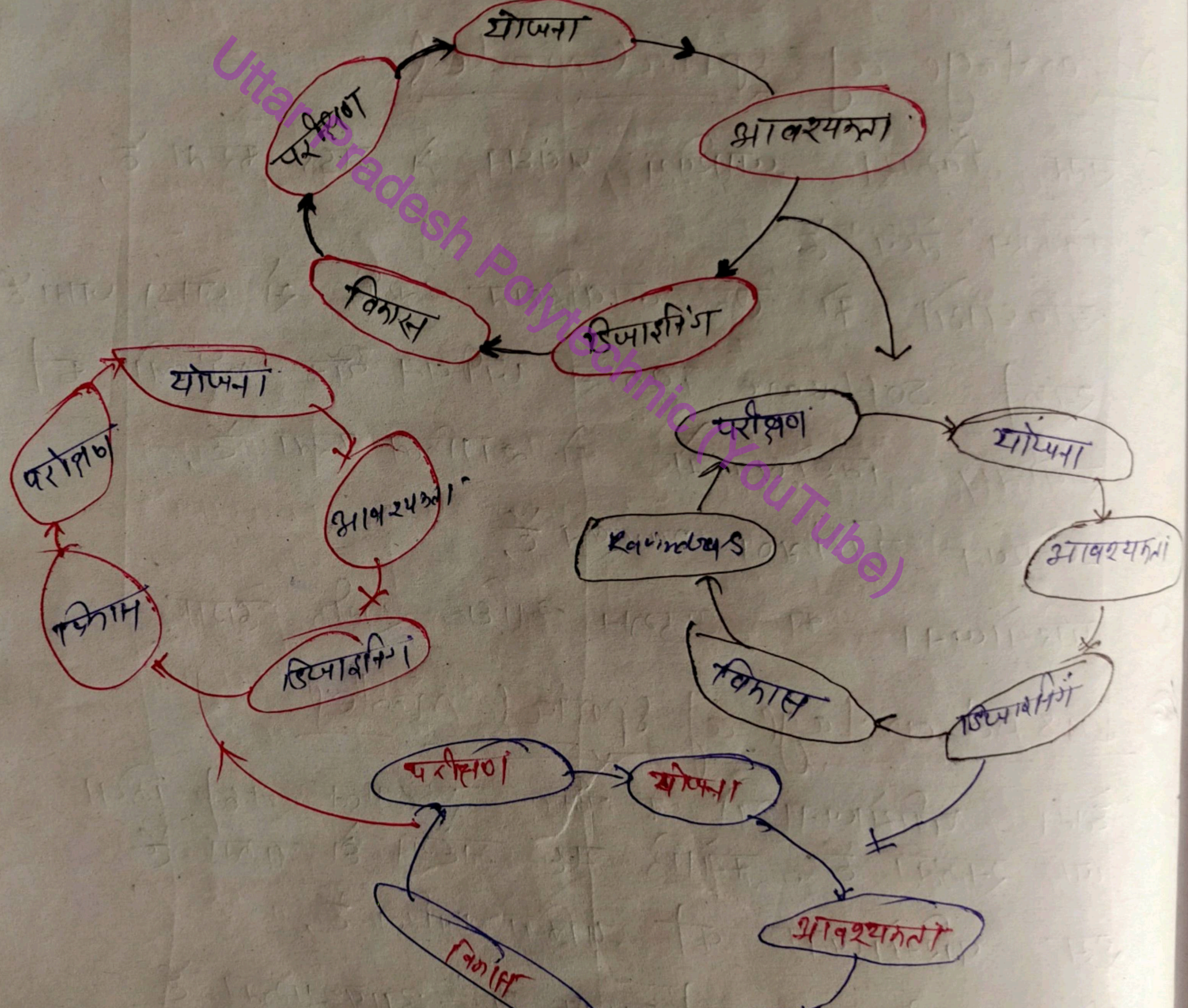
Dis Advantage of spiral model

1. छोटी परियोजनाओं के लिए इस्तेमाल नहीं किया जा सकता है, क्योंकि यह महंगा हो सकता है,
2. इस परियोजना की प्रक्रिया जटिल है,
3. प्रलेखन की एक बड़ी मात्रा मे आवश्यकता है,
4. परियोजना को चलाने के लिए विशेषज्ञ टीम आवश्यक है
5. कम जोखिम, वाली परियोजनाओं के लिए उपयुक्त नहीं है

Uttar Pradesh Polytechnic (YouTube)

Agile methodology (एजाइल मॉडल)

यह कार्यप्रणाली परियोजना के software विकास जीवन चक्र में विकास और परीक्षण का निरंतर प्रक्रिया की पुनरावृत्ति है।
इस तरीके में विकास और परीक्षण गतिविधियाँ एक साथ चल रही है।



Uttar Pradesh Polytechnic (YouTube)

[एजाइल मॉडल]

प्रत्येक प्रगतिशील भाग

1. आवश्यकता इकट्ठा करना और विश्लेषण
2. आवश्यकताओं की डिजाइनिंग
3. पुनरावृत्ति
4. परीक्षण
5. परिनियोजन / तैनाती
6. प्रतिक्रिया

एजाइल कार्यप्रणाली की विभिन्न विधियाँ

- स्क्रम (Scrum)
- क्रिस्टल तरीके
- DSDM (गतिशील software विकास विधी)
- सुविधा संचालित विकास
- Lean (पतला) सॉफ्टवेयर (विकास)
- एक्सट्रीम प्रोग्रामिंग (XP)

Uttar Pradesh Polytechnic (YouTube)

Agile model / Advantage -

- यह अन्य सभी तरीकों के बीच बहुत प्रभावी है,
- यह सॉफ्टवेयर डेवलपमेंट के लिए पैमाने पर आधारित अप्रोच प्रदान करता है
- अंतिम उपयोगकर्ताओं के लिए टीम work और और प्रशिक्षण को बढ़ावा देता है।
- उत्पाद तेजी और लगातार वितरण किया जाता है,

(सॉफ्टवेयर योजना)

Software Planning

Lesson ⇓ 3

Software planning ⇒ किसी software project के बारे शुरू से अंत तक लेकर योजना बनाना जिसमें उसके आकार (size) का अनुमान उससे आने वाली लागत का अनुमान, परियोजना (Project) की अवधि का अनुमान और उसके लिए क्या-क्या प्रयासों की आवश्यकताओं शामिल है, Software planning कहते है,

Uttar Pradesh Polytechnic (YouTube)

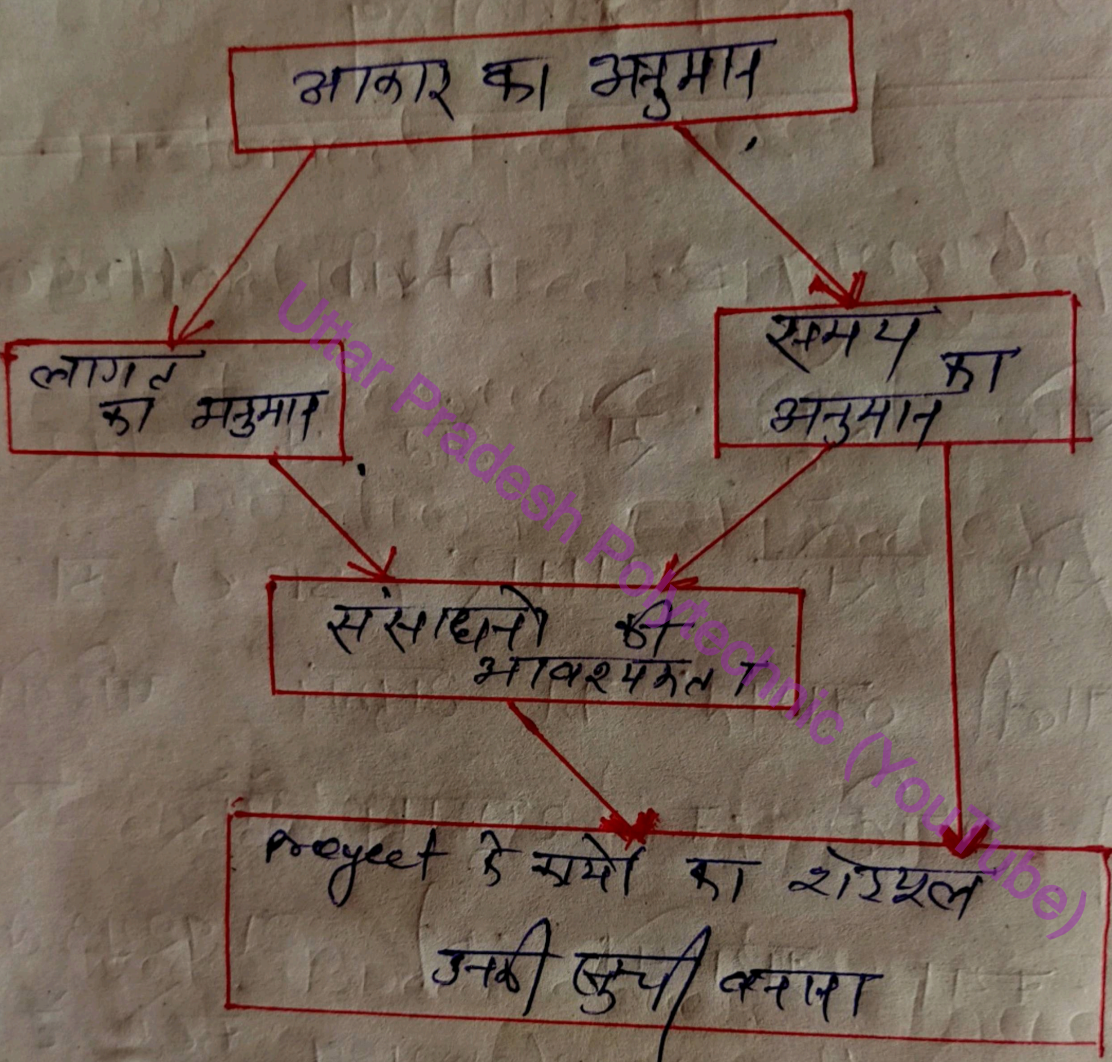
Project planning की कुछ गतिविधियाँ

1→ परियोजना (Project) के आकार का अनुमान

2→ परियोजना की लागत का अनुमान

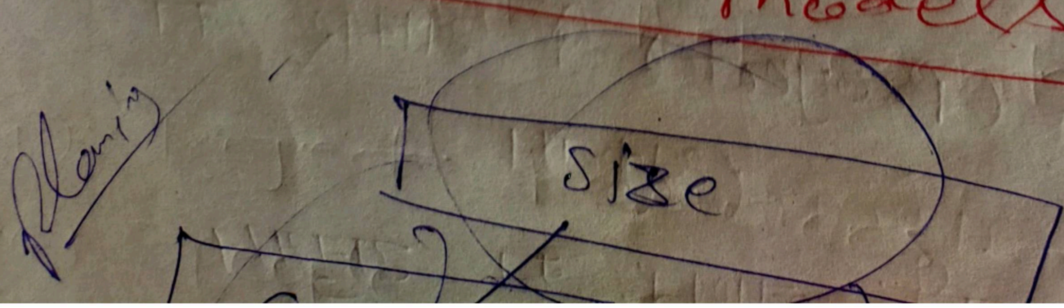
3→ प्रोजेक्ट की अवधि

4→ गुणवत्ता निर्धारण (Quality Determination)



Uttar Pradesh Polytechnic (YouTube)

Software Engineering का Planing models



② SPMP

SPMP Stand For [Software project management plan]

Software project management (SPM) Software project की योजना बनाने और उसे आगे बढाने का एक उचित तरीका है, यह project management का एक हिस्सा होता है,

यह project management का एक हिस्सा होता है,

जिससे Software projects को plan, implement monitor तथा control किया जाता है/

1) Introduction

(a) उद्देश्य (objectives)

(b) प्रमुख कार्य (major Functions)

(c) निष्पादन मुद्दे (performance and Technical

(d) प्रबंधन और तकनीकी बाधाएँ (management and Technical constraint's)

Uttar Pradesh Polytechnic (YouTube)

(2) Project Estimates: (अनुमान)

(a) Historical Data used

(b) Estimation Techniques used

(c) Effort, Resource, Cost, and project Duration Estimates

(3) Schedule: (अनुसूची)

(a) Work Breakdown Structure

(b) Task Network Representation

(c) Gantt chart Representation

(d) PERT Chart Representation

(4) Project Resources (संसाधन)

(a) People

(b) Hardware and Software

(c) Special Resources

(5) Staff organization (कर्मचारी संगठन)

(a) Team Structure

(b) management Reporting

(6) Risk management Plan

(7) Project Tracking and Control plan

(8) miscellaneous plans

Uttar Pradesh Polytechnic (YouTube)

Important

LOC (LINES of Code)

LOC ⇒ यह मॉडल स्थैतिक तथा single valued होता है, जो कि software development effort तथा cost को Function के program size की तरह compute करता है, तथा program size जो है, वह अनुमानित (estimated) lines of code (LOC) में व्यक्त होता है, सामान्यतया इस मॉडल का प्रयोग छोटे तथा मध्यम आकार के software project में किया जाता है।

Uttar Pradesh Polytechnic (YouTube)

Feature of LOC

1. यह software program के source code के Text से लाइनों की संख्या की गणना करता है,
2. कोड में टिप्पणी comment करने वाली लाइनों को अनदेखा किया जाता है, और हेडर फाइल
3. LOC का अनुमान प्रोग्राम के आखिर में तो आसानी से लगाया जा सकता है परन्तु शुरुआत में गणना करना मुश्किल है,
4. यह समस्या के आकार का संख्यात्मक रूप मान देता है।

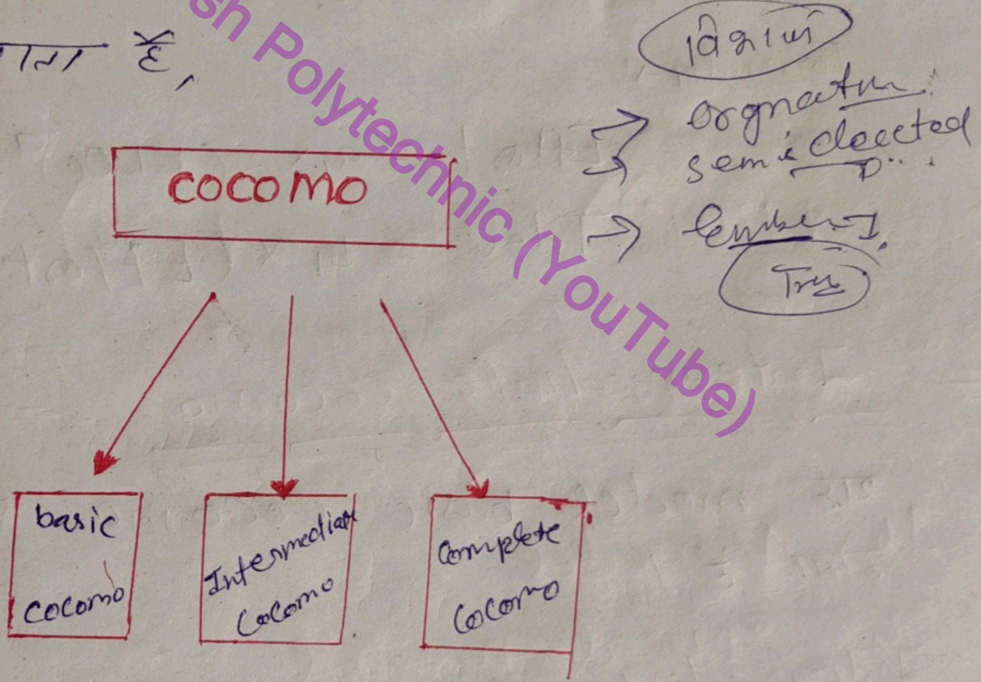
COCOMO (कोकोमो)

COCOMO stand for - Constructive Cost model है, [illegible].

1981 in Release by BOEHM (बोहम)

"यह एक cost estimation मॉडल है, तथा इसका प्रयोग software packages की cost को evaluate करने लिए किया जाता है,

इस model में effort equation का प्रयोग एक project को विकसित करने के लिए person-month संख्या की जरूरत को estimate करने के लिए किया जाता है,



1. Basic Cocomo model
2. Intermediate Cocomo model
3. Complete Cocomo model

Uttar Pradesh Polytechnic (YouTube)

1→ Basic Cocomo model

यह एक स्टैटिक तथा single valued होता है, जो कि software Development effort तथा cost को Function के program size की तरह compute करता है,
तथा program size जो है, वह अनुमानित (estimated) lines of code (LOC) मे व्यक्त करता है,

इस model मे cost को estimate करने के लिए

Formula →

$$\text{Effort} = a_1 \times (\text{KLOC})^{a_2} \text{ pm}$$

$$\text{Tdev} = b_1 \times (\text{Effort})^{b_2} \text{ months}$$

Uttar Pradesh Polytechnic (YouTube)

2→ Intermediate Cocomo

यह model basic Cocomo का extension होता है।

Intermediate Cocomo software development effort को program size के Function तथा cost drivers के समूह की तरह compute करता है।

cost drivers project में लगे समय तथा effort को निर्धारित करता है,

" यह model basic model से बेहतर परिणाम देता है, क्योंकि इसमे cost drivers का प्रयोग किया जाता है,

3→ Complete Cocomo

यह model Intermediate cocomo को extension होता है।

यह model Intermediate model से अलग होता है, क्योंकि यह project के प्रत्येक phase के लिए effort multipliers का प्रयोग करता है।

"complete cocomo मे प्रत्येक subsystem की cost को अलग-2 estimate किया जाता है, इस विधी के कारण Error (त्रुटी) कम होती है,

Basic तथा Intermediate cocomo की कमी यह है कि यह software project को single homogeneous entity की तरह consider करता है। इसी कमी को complete cocomo दूर कर देता है

Uttar Pradesh Polytechnic (YouTube)

Cocomo model कि कुछ विशेषताएँ

① Cocomo एक Constructive Cost Estimation model) है,

② यह एक ऐसा model है, जो project Parameters के द्वारा सटीक आकार को ज्ञात करने में help करते है,

③ Basic Cocomo model ऐसा model जोकि Constants होता है, तथा project development में effort व cost के सभी Function के size को manage करता है।

④ BOEHM के अनुसार Cost को estimate करने के लिए तीन Stages का प्रयोग किया जाता है।

1. V model
2. Spiral model
3. Sdlc model.

Uttar Pradesh Polytechnic (YouTube)

(Requirement Analysis and Specification)

Lesson - 4

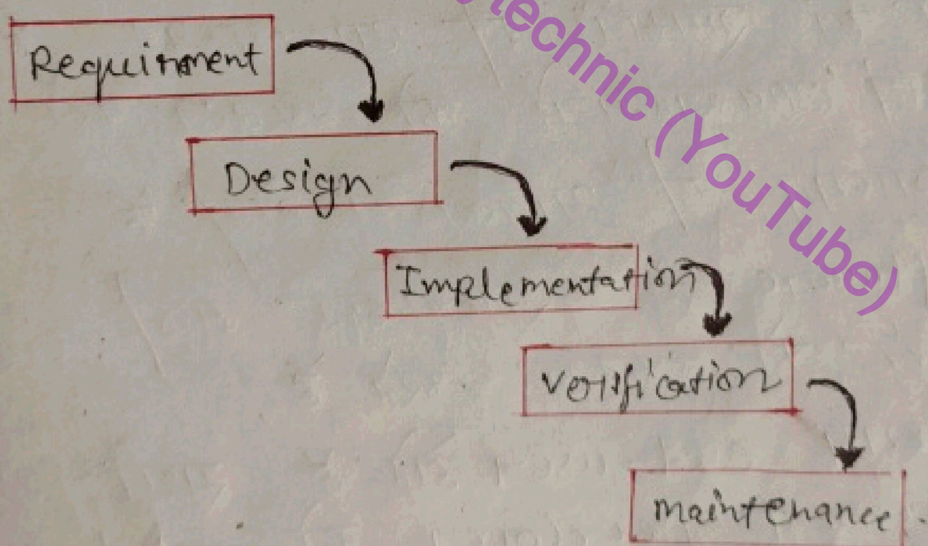
आवश्यकता विश्लेषण और विनिर्देश

Topic ⇒ ↳ Requirement gathering and Analysis, ✓

↳ Software Requirement Specifications (SRS) ✓

↳ Characteristics of good SRS

Software Development D/G



- आवश्यक विश्लेषण में आने वाली समस्याओ को कुछ इस प्रकार जानते है।

1. Data redundancy / repetition.
2. Accuracy
3. Timeliness
4. Error / bug
5. Ambiguity.

Uttar Pradesh Polytechnic (YouTube)

Requirement gathering — V.Imp

Requirement gathering में Gathering का अर्थ होता है, इकट्ठा करना,
अतः requirement gathering एक ऐसी प्रक्रिया है, जिसमे software की आवश्यकताओं को इकट्ठा करने के लिए ग्राहकों (customers) stake, holders तथा end users के साथ Interact किया जाता है।

किसी भी software को बनाने के लिए सबसे पहले आवश्यकताओं (Requirement's) को जानने एवं समझने की आवश्यकता होती है, इन सभी आवश्यकताओं को इकट्ठा करने के लिए एक माध्यम की आवश्यकता होती है जो software से सम्बन्धित सभी आवश्यक जानकारी को इकट्ठा करते हैं,

ये माध्यम end-user होते हैं सभी आवश्यक requirement Developer को end-use द्वारा दी जाती है।

end-use एक Individual person होते हैं, जो system या संगठन से direct या किसी और रूप से जुड़े होते हैं।

Ex- manager, Tester, software Engineering Developer

Uttar Pradesh Polytechnic (YouTube)

Requirement gathering की विधिया

1) **Interview** ⇒ इस विधि द्वारा software से ग्राहक (customer) की क्या उम्मीदे है, इस विधि मे end-user और steak holder से Interview लिया जाता है,

2) **Scenario (पटकथा)** ⇒ Scenario एक step by step प्रक्रिया होती है, Scenario मे एक श्रृंखला (series) के अनुसार चला जाता है, कि किस scenario मे यह देखा जाता है, कि किस प्रकार की और कितनी I/P की आवश्यकता होगी, इसमे user की आवश्यकता के अनुसार step by step सुचना information collect किया जाता है।

3) **Questionnaire** ⇒ इसमे कम समय और कम कीमत मे software से सम्बन्धित महत्वपूर्ण सूचना को इकट्ठा करते है। इस विधी मे यह होता है, कि सुचनाओ को अलग-2 देते है। क्योकी सबका अलग-2 क्षेत्र होता है। अपने-अपने department की पूर्ण जानकारी देते है।

4) On Side observation (प्रत्यक्ष अवलोकन)

5) (Prototype) मूलरूप

Uttar Pradesh Polytechnic (YouTube)

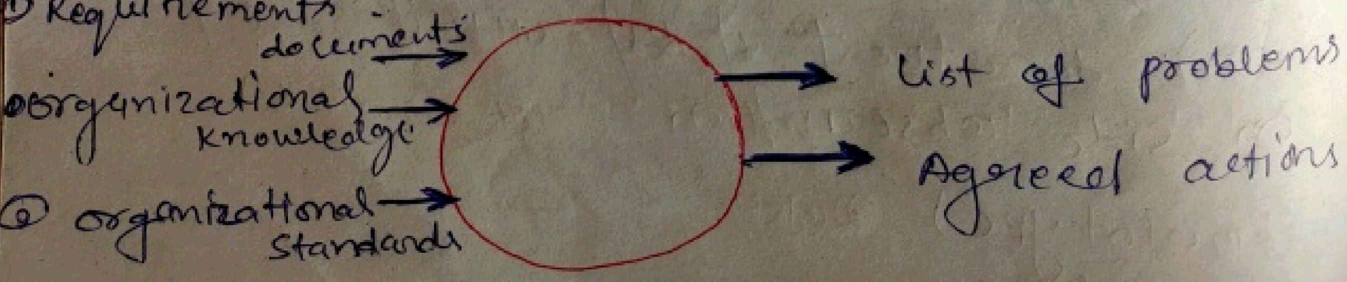
Requirement Analysis

Software designing तथा Development में किसी भी प्रकार की कोई समस्या न आए इसके लिए requirement analysis अर्थात् आवश्यकता विश्लेषण का प्रयोग किया जाता है।
इसे requirement validation भी कहते है।

"किसी भी संगठन में, प्रत्येक नए उत्पाद (product) या सेवा (service) को व्यवसाय (business) की आवश्यकता के जवाब (Response) से बनाया जाता है, हालांकि विकास (development) पर समय और संसाधन खर्च करने के बावजूद, आवश्यक उत्पाद (Required product) और अंतिम उत्पाद (Final product) के बीच एक बेमेल हो सकता है।
इसलिए, भविष्य में बड़ी समस्याओं से बचने के लिए किसी भी परियोजना के प्रारंभिक चरणों में एक केंद्रित (Focused) और विस्तृत (detailed) आवश्यकताओं के विश्लेषण की आवश्यकता है।

Uttar Pradesh Polytechnic (YouTube)

Requirement Analysis का Input / output



Very Imp

SRS
(Software Requirements Specifications)

Software requirement specifications जो है, वह किसी सिस्टम या Software Application की जरूरतों का एक पूरा Document का विवरण होता है।

SRS एक Document होता है, जो कि यह Describe करता है, कि Software के Features क्या होंगे तथा उसका behaviour क्या होगा अर्थात् वह किस प्रकार परफार्म करेगा.

"SRS का फायदा यह है, कि इससे developers को Software को विकसीत करने में कम समय तथा मेहनत लगती है,,"

" SRS जो है, वह Software के layout की तरह होता है, जिसकी यूजर समीक्षा कर सकता है, तथा देख सकता है, कि वह (SRS) उसकी जरूरतों के हिसाब से बना है, या नहीं,,"

Uttar Pradesh Polytechnic (YouTube)

Characteristics of SRS

1:- **Complete (संपूर्ण)** ⇒ SRS जो है वह संपूर्ण होना चाहिए। जो भी ~~SRS की~~ software की जरूरत हो SRS में लिखा होना चाहिए।

(2)- **Correct (उचित)** ⇒ यह सही होना चाहिए अर्थात् जो customer की जरूरतें हैं, उसके हिसाब (गो) होना चाहिए।

(3) **Clear (स्पष्ट)** ⇒ यह स्पष्ट होना चाहिए, software की आवश्यकताओं को स्पष्ट रूप से वर्णित (declared) होना चाहिए।

(4) **accurate (परिशुद्ध)**, इसमें accuracy होनी चाहिए, अगर यह ही accurate नहीं होगा, तो software का निर्माण नहीं हो सकता।

Uttar Pradesh Polytechnic (YouTube)

(5) consistent (निरंतर)

(6) verifiable (सत्यापनीय) ⇒ यह verifiable (सत्यापन योग्य) होना चाहिए, जो Experts तथा testers होते हैं उनके द्वारा requirements को verify किया जाता है,

7) modifiable (परिवर्त्य) ⇒

SRS का जो document होता है, उसमें specify की गयी सारी रिक्वायरमेंट्स मॉडिफाई करने योग्य होनी चाहिये. यह तभी हो सकता है, जब SRS का स्ट्रक्चर संतुलित हो.

Uttar Pradesh Polytechnic (YouTube)

8) Traceable

9) testable (परिक्षण योग्य)

यह testable होना चाहिए अर्थात इसे किसी भी प्रकार से Test करने के योग्य होना चाहिये।

10) unambigous (अमीश्रित) ⇒ कहने का मतलब केवल एक ही Interpretation हो (व्याख्या) हो।

5 Software Design and Implementation

Topic → Characteristics and Features of good Software Design Cohesion and cupling,
↳ Software design Approaches-
↳ Function oriented Design (Data Flow diagrams, Data dictionary, Decision Trees and tables)
↳ object oriented Design, Structured coding - Techniques, coding Styles, and documentation

Uttar Pradesh Polytechnic (YouTube)

COHESION

⇒ cohesion एक माप है, जो कि एक मॉड्यूल के अन्दर elements की intradependability की degree को परिभाषित करता है, अर्थात् cohesion मे यह देखा जाता है, कि एक module के element's एक-दूसरे से कितने करीब से सम्बन्धित हैं।
जितना ~~ज्यादा~~ ज्यादा cohesion होता है, उतना उच्च Software डिजाइन हो जाता है।

(A) Types of Cohesion

1. Coincidental Cohesion
2. Logical Cohesion
3. Temporal Cohesion
4. Procedural Cohesion
5. Communicational Cohesion
6. Sequential Cohesion
7. Functional Cohesion

(1) Coincidental Cohesion ⇒

यह एक Cohesion सबसे निम्न स्तर का cohesion होता है। यह तब घटित होता है, जब module के Elements के मध्य कोई सम्बन्ध नही होता या जब program को छोटे modules मे तोड़ दिया जाता है।

(2) Logical Cohesion ⇒

एक modules मे logical cohesion तब होता है जब एक modules के मध्य logical Relation होता है। तथा Elements एक ही logical class मे रखे गये Function को प्रकार्य करते है।

Uttar Pradesh Polytechnic (YouTube)

Temporal Cohesion ⇒

Temporal Cohesion लॉजिकल Cohesion से उच्च स्तर का होता है।
क्योंकि इसमें element एक ही समय पर process होता है,

4. Procedural Cohesion. ⇒

जब मॉड्यूल के Element's एक साथ समूह में होते हैं, तो वह एक Task को पूरा करने के लिए एक क्रम में निष्पादित (execute) होते हैं

5. Communicational Cohesion ⇒

6. Sequential Cohesion ⇒

जब मॉड्यूल के Element एक साथ एक समूह (group) में होते हैं, क्योंकि Element's का output दूसरे element के लिए Input की तरह होता है,

7. Functional Cohesion ⇒

यह Cohesion का सबसे उच्चतम स्तर होता है, module के सभी element एक Function को परफॉर्म करने से सम्बन्धित हैं।

Uttar Pradesh Polytechnic (YouTube)

Coupling ⇒ coupling की माप (measurement) है, जो एक program के मॉड्यूल के मध्य Interdependability के स्तर को define करता है।

अर्थात्

यह एक प्रकार की माप है, जो कि यह define करती है, कि software के components एक-दूसरे पर किस प्रकार निर्भर है, तथा Interact करते coupling जितनी कम होगी software Design उतना ही बेहतर होगा।

Uttar Pradesh Polytechnic (YouTube)

1. Content Coupling ⇒

Content coupling सबसे उच्च स्तर की coupling होती है, यह तब होती है, जब कोई मॉड्यूल किसी दूसरे module की आन्तरिक क्रियाविधि पर निर्भर होता है।

2. Common Coupling ⇒ इसे global coupling भी कहते है।

यह तब घटित होता है, जब समान global data दो मॉड्यूल्स के द्वारा share किया जाता है

3. External Coupling ⇒ इस प्रकार की coupling तब घटित होता है जब External data format

द्वारा Communication protocol को दो मॉड्यूल्स के द्वारा share किया जाता है।

4. Control Coupling ⇒

एक module दूसरे module के Flow को नियंत्रित करता है।

5 ⇒ Stamp Coupling ⇒

6 ⇒ Data Coupling ⇒

दो मॉड्यूल्स के मध्य डाटा कपलिंग तब होगी जब Data केवल उनके मध्य पास होगा।

Uttar Pradesh Polytechnic (YouTube)

→ (1) Content
(2) common Coupling
(3) External
(4) Control

DFD (Data Flow Diagram)

Data Flow Diagram Information system से गुजरने वाले Data के Flow का एक ग्राफिकल प्रस्तुतिकरण है।
यह Data के आने-जाने एवं Store होने तक की सूचना रखता है।
Data Flow Diagram का प्रयोग software system के ~~ओवर~~ ओवरविव को बनाने के लिए किया जाता है।

Uttar Pradesh Polytechnic (YouTube)

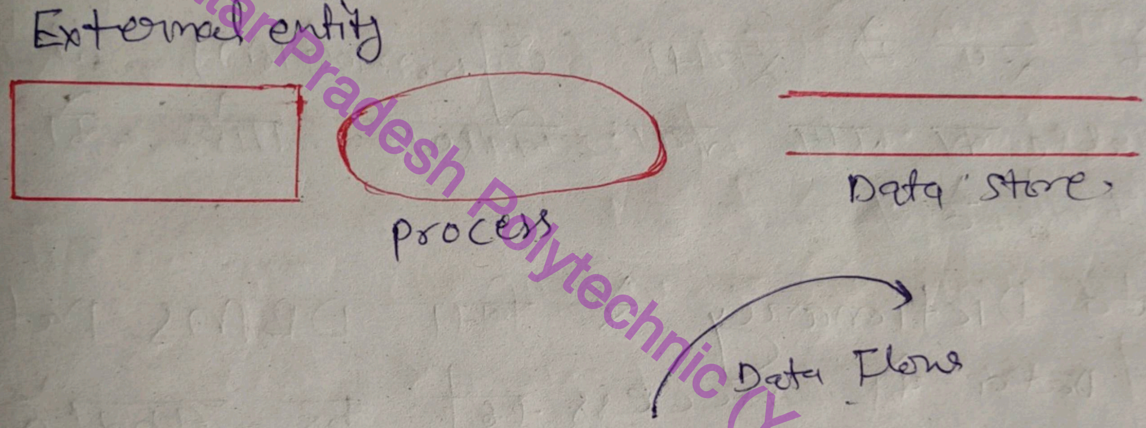
DFD के प्रकार

1. DFD Logical DFD ⇒
2. DFD Physical DFD

1. DFD Logical DFD ⇒ Logical DFD बिजनेस Activitie पर केन्द्रित रहता है, अर्थात् यह System से Data के Flow तथा System के Processes पर केन्द्रित रहता है।

2) **Physical DFD** → Physical DFD इस बात पर केन्द्रित रहता है, कि वास्तव में Data Flow system किस प्रकार कार्यान्वित हुआ है,

Data Flow Diagram का सबसे पहला प्रयोग 1970 के दशक में किया था, तथा इसे larry Constantine तथा Ed yourdon ने वर्णित किया था।



Uttar Pradesh Polytechnic (YouTube)

3) DFD के Components

1) **Entities** → Data के source तथा destination को entities कहते है।

2) **Data Storage** ⇒ Data storage होता है।

3) **process** ⇒

4) **Data Flow** ⇒

Data Dictionary

Data dictionary DBMS में Data Dictionary एक File या फाइलों का समूह होती है, जो कि Database के metadata को store करती है।

Data Dictionary Database के वास्तविक Data को एकत्र नहीं करती है। बल्कि यह सिर्फ Data को मैनेज करने के लिए बनी जाती के रूप में सूचना (information) जैसे, Table का नाम तथा विवरण आदि को संग्रह करती है।

Data Dictionary के बिना DBMS Database से Data को Access नहीं कर सकता है।

Data Dictionary का प्रयोग Database operation, Data integrity तथा शुद्धता (accuracy) को नियन्त्रित करने के लिए किया जाता है।

Type of Data Dictionary

1. Active Data Dictionary
2. Passive Data Dictionary

Uttar Pradesh Polytechnic (YouTube)

ODD (Object Oriented design)

Object Oriented Design (OOD) एक प्रक्रिया है, जिसमें object oriented विधियों का प्रयोग गणना प्रणाली (Computing system) या Application को create करने के लिए किया जाता है।

OOD मे एक system को object's के collection के रूप में देखा जाता है, OOD (Object oriented programing) की प्रक्रिया का एक भाग है।

Terms of object oriented Design

1. **Object's** ⇒ Object एक class का Instance होता है, यह (properties) s contain किये रहता है, Ex- person, Bank, company etc.

2. **Class** ⇒ क्लास एक Entity होती है, जो यह निर्धारित करता है, कि object किस प्रकार व्यवहार करेगा,

3. messages
4. Abstraction
5. Encapsulation
6. Inheritance → ① Sub Class ② Superclass
7. polymorphism

Uttar Pradesh Polytechnic (YouTube)

Advantage of Decision Tree

1. इसके लिए किसी भी डोमेन ज्ञान की आवश्यकता नहीं होती है।

2. Decision Tree के सीखने और वर्गीकरण के चरण सरल तेज होते हैं।

Decision Table

यह Tool प्रयोग परीक्षण एवं आवश्यकताओं के प्रबन्धन के लिए तैयार किया गया है। इससे यह देखना आसान हो जाता है, कि सभी सम्भावित संयोजनों पर विचार किया गया है, या किसी संयोजन को छोड़ा तो नहीं गया है, यदि प्रक्रिया जटिल हो तो Conditional निर्णय लेते होते हैं।

Uttar Pradesh Polytechnic (YouTube)

Decision Table Four Types

1. Condition stub

2. Condition Entry

3. Action stub

4. Action Entry

Important Structured programing -

Structured programing मे, हम पूरे कार्यक्रम को छोटे module मे उप-विभाजित करते है, ताकि कार्यक्रम को समझना आसान हो
Structured programing का उद्देश्य computer program के माध्यम से नियन्त्रण प्रवाह को रैखिक करना है,
ताकि निष्पादन : अनुक्रम उस क्रम का अनुसरण करे जिससे कोड लिखा गया है।
कार्यक्रम की गतिशील संरचना कार्यक्रम की स्थिर संरचना से मिलती जुलती है।
यह कार्यक्रम की पठनीयता, परीक्षणशीलता और परिवर्तनशीलता को बढ़ाता है।

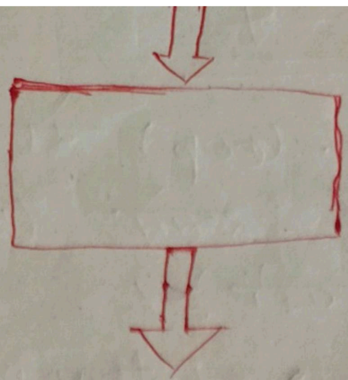
(2) Structured programing नियम & Code आरेख

संरचित प्रोग्रामिंग का प्रथम नियम

Flow-charting की स्थित में, एकल प्रविष्टि बिन्दु और एकल निकास बिन्दु वाले बॉक्स को संरचित किया जाता है,

Uttar Pradesh Polytechnic (YouTube)

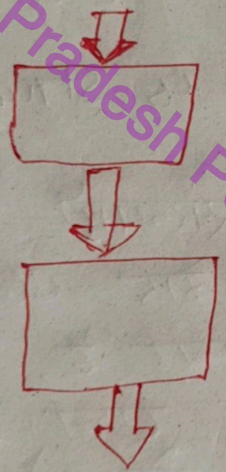
{structured programming First Rule's-1}



{code block is structured}

संरचना का दूसरा नियम ⇒ अनुक्रम Block का क्रम सही है, यदि प्रत्येक Block की निकास स्थितियाँ निम्न Block की प्रवेश स्थितियों से मेल खाती हैं,

Rule's-2

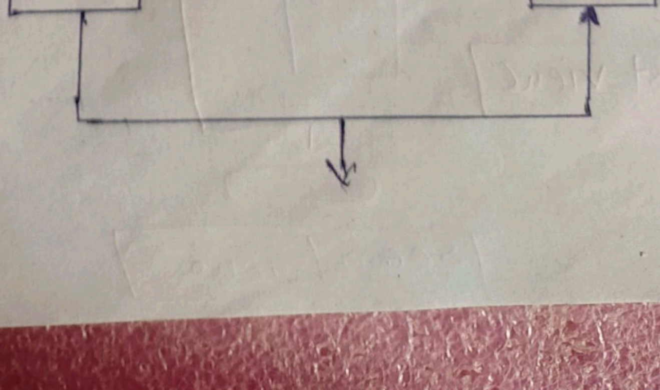


{A sequence of code blocked is structured}

Uttar Pradesh Polytechnic (YouTube)

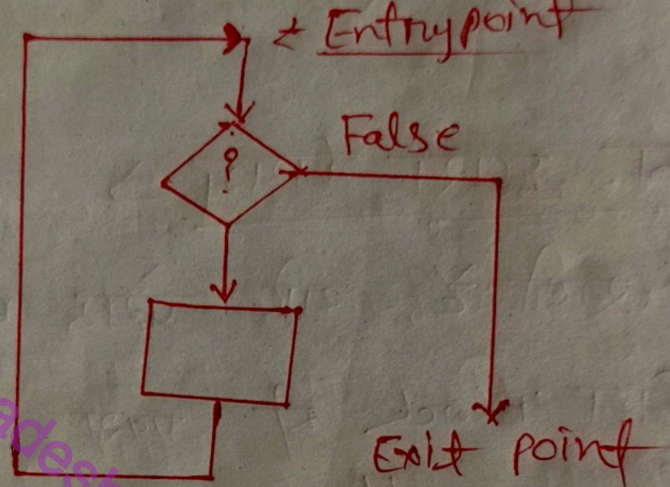
संरचना का तीसरा नियम ⇒ दो code Block का विकल्प संरचित है,

(An Alternation of code blocks is structured)



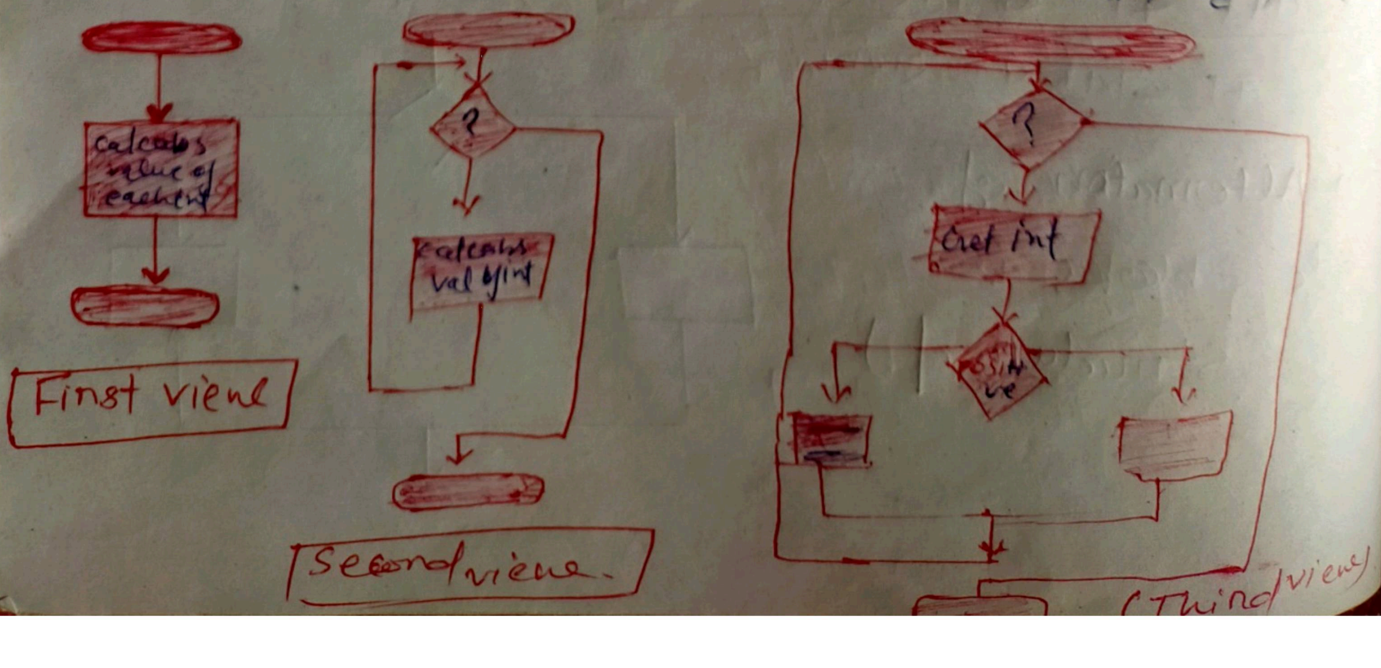
संरचित नियम - 4

Interation (while loop) को सही रूप में व्यवस्थित किया जाता है, इसमें एक प्रवेश बिंदु और निकास बिंदु होता है।



(Iteration of code blocks is structured)

संरचित नियम - 5 नेस्टेड संरचनाएँ फ्लोचार्टिंग परिस्थितियों में, किसी भी कोड Block को किसी संरचना में फैलाया जा सकता है, यदि फ्लोचार्ट का एक भाग है, जिसमें एक एकल प्रवेश बिंदु और एक निकास बिंदु है तो इसे एकल code Block के रूप में संक्षेपित किया जा सकता है।

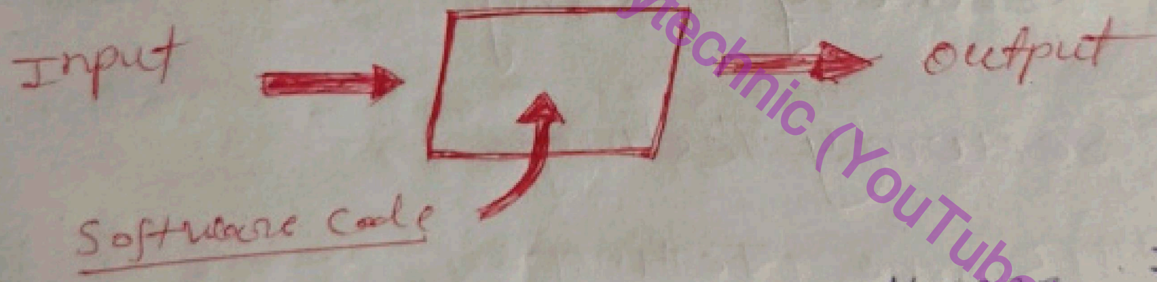


Uttar Pradesh Polytechnic (YouTube)

Lesson-6 सॉफ्टवेयर परीक्षण (Software Testing)

Testing ⇒ Testing software का एक प्रक्रिया है, जिसके अन्तर्गत system या system उपकरणों की जाँच की जाती है,
इस परीक्षण का उद्देश्य यह है, की सिस्टम कम्पनी द्वारा निर्धारित मानको के अनुसार तैयार किया गया हो,

Black box Testing



Uttar Pradesh Polytechnic (YouTube)

विक्रेता कंपनी द्वारा तैयार software को ग्राहक के जरूरतो के अनुसार विकसित किया जाता है, परिस्थितियो मे तयार की और ग्राहक जिन परिस्थितियो को लैब मे तयार कर software को परखता है।
अर्थात software की Testing करता है।

Types of Software Testing

1. Unit Testing
2. Integration testing → Top down, bottom up, sandwich, big-bang
3. Alpha Testing
4. Beta Testing
5. Stress Testing
6. Recovery Testing
7. Security Testing
8. Smoke Testing
9. Regression Testing
10. System testing
11. Performance Testing.
12. Ad-hoc Testing.

Uttar Pradesh Polytechnic (YouTube)

"V" model's

V-model's को verification तथा Validation (V&V) models भी कहते है।
V-model को verification एक SDLC model है, जहाँ process को V shape मे execute किया जाता है.

V-model softwareDevelopment Life cycle के प्रत्येक phase के लिए उपयोगी है,

V-model वह vecterfall model extension है, तथा यह निर्धारित करता है, कि प्रत्येक Development phase से जुड़ा हुआ है,

Uttar Pradesh Polytechnic (YouTube)

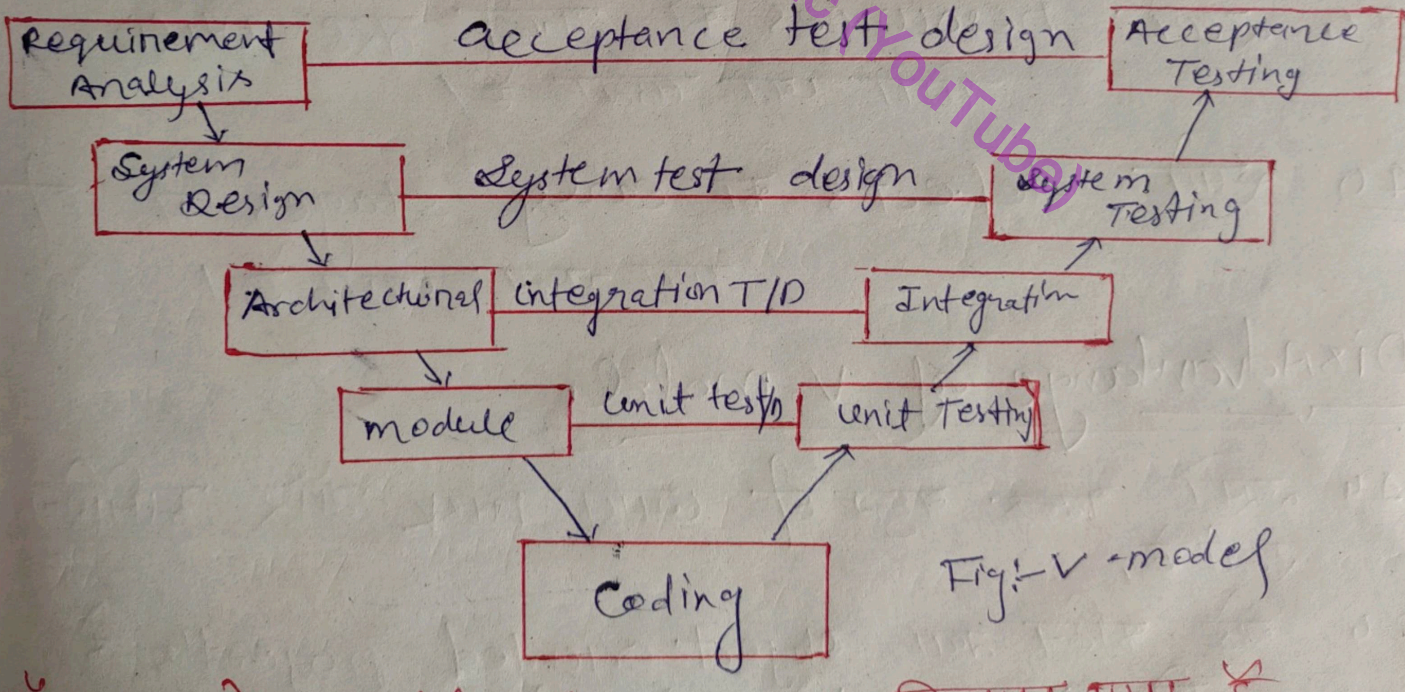


Fig:- V-model

दाँये पक्ष में verification phases दिखाया गया है, तथा बाँए पक्ष में validation phases प्रदर्शित किया गया है।

Verification phases	Validation phase
1→ Requirement Analysis	(1) Unit Testing
2→ System Design	(2) Integration Testing
3→ Architectural Design	(3) System Testing
4→ Module Design	(4) Acceptance Testing
5→ Coding phase	(5)

Advantage of V-models

1→ यह बहुत ही ज्यादा disciplined model है, इसमे एक समय में एक phase को पुरा कर सकते है

2→ इसमे planning तथा Tests डिजाइनिंग, Coding से पहले हो जाती है, जिससे समय बचता है

3→ इसमें resources को अच्छी तरह utilize किया जा सकता है।

4→ इसमे defect's का early stages मे ही पता चल जाता है।

DisAdvantage of V-models

1→ इसमे बहुत ही ज्यादा risk और अनिश्चितता रहती है,

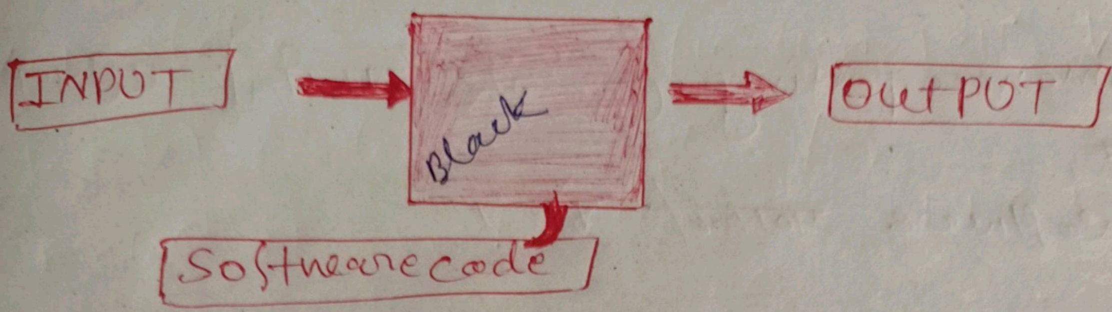
2→ बड़े कठिन तथा object oriented project's के लिए यह model सही नहीं है,

3→ यह Flexible model नहीं है।

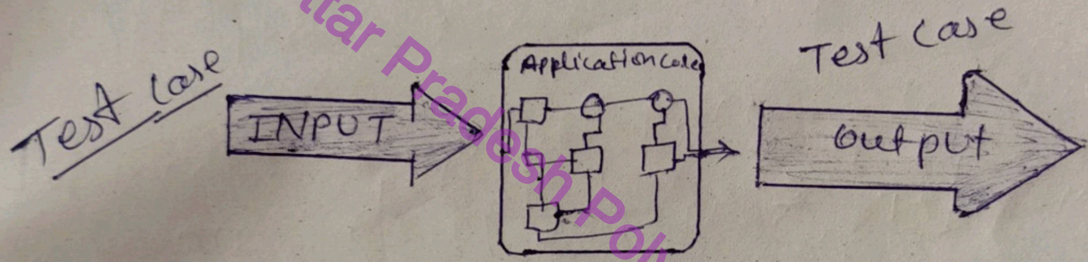
4→ इसमें अगर बीच मे कोई बदलाव कर दिया तो Test Document's को भी change करना पड़ता है।

Uttar Pradesh Polytechnic (YouTube)

(A) Black Box Testing (BBT)



(B) White Box Testing – (WBT)



Uttar Pradesh Polytechnic (YouTube)

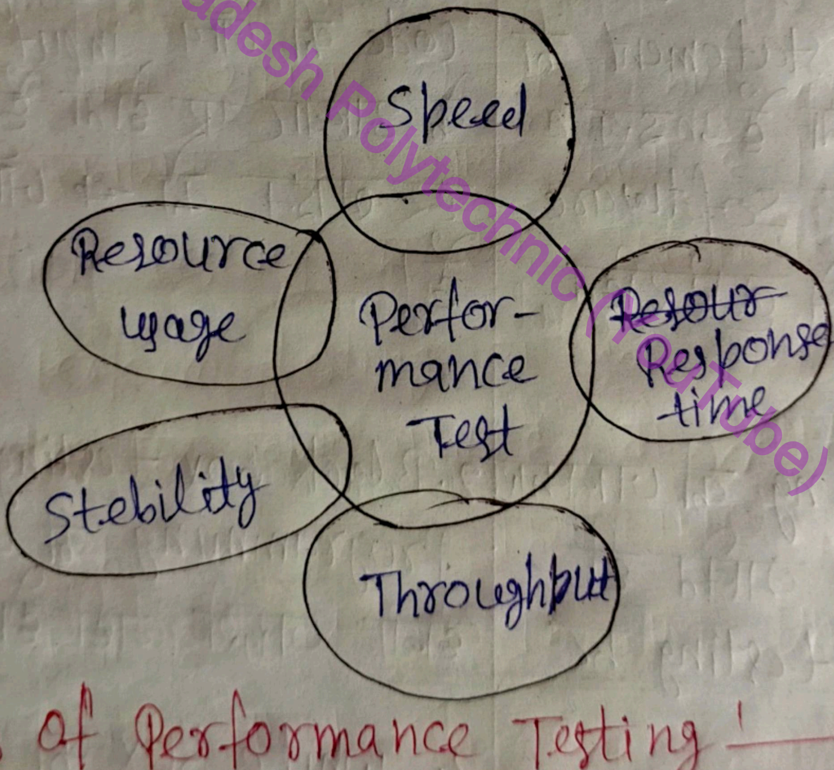
White Box Testing (WBT)	Black Box Testing (BBT)
1) WBT मे software के internal code की जाँच की जाती है,	(1) BBT मे केवल software के INPUT और output की जाँच की जाती है,
2) WBT के दौरान, सोर्स कोड मे लिखित हर statement, loop की जाँच, Test केसेस के आधार पर की जाती है,	(2) Black box Testing मे Test केसेस के आधार पर केवल Input और output की जाती है।
(3) programing की जानकारी वाले विशेषज्ञ व्यक्ति ही व्हाइट बॉक्स टेस्टिंग कर सकते है।	(3) Black box Testing के लिए programing की जानकारी होना जरूरी नही होता है।

Performance Testing :-

Performance testing एक गैर-तकनीकी Testing पद्धति है। जिसमें सॉफ्टवेयर प्रणाली को स्थिरता, सुचारु रूप से कार्य करने का तरीका जैसे अन्य मानकों को अलग-अलग workload पर जाँचा जाता है।

Performance testing का मुख्य उद्देश्य software की विश्वसनीयता, स्थिरता, मापनीयता और उपलब्ध संसाधनों का कम उपयोग करने की क्षमता को परखना है।

Uttar Pradesh Polytechnic (YouTube)



Types of Performance Testing :-

1. Load Testing :-

2. Stress Testing

Ex- Web Server की stress Testing shell script (शेल script, एक script है, जिसे Operating system के शेल या comand interpreter के लिए इस्तेमाल किया जाता है।)

Unit testing :-

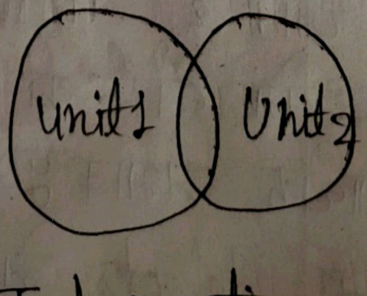
Unit testing Software testing की एक विधि है जिसमें Software application के सबसे छोटे unit (भाग) की Testing किया जाता है इस Testing का मुख्य उद्देश्य यह सुनिश्चित करना है कि Software की प्रत्येक unit का source code सही है, तथा इसे प्रयोग किया जा सकता है। पहले adhoc tools का प्रयोग unit को Test करने के लिए किया जाता है परन्तु आजकल unit Test करने के लिए frame (Java frame work, dot net frame work तथा PHP Frame work आदि) का प्रयोग किया जाता है।

Unit testing Developers के द्वारा लिखे तथा निर्दिष्ट किये जाते है तथा इसे करने के लिए White box testing विधी का प्रयोग किया जाता है। Unit testing में अधिक समय लगता है। तथा इसमें धैर्य की आवश्यकता होती है यह परीक्षण बहुत ही अधिक प्रभाव पूर्ण है, क्योंकि इसके प्रयोग के द्वारा अधिकतर defects को identify कर

Uttar Pradesh Polytechnic (YouTube)

लिया जाता है। यह परीक्षण सामान्यतया integration testing के ~~बाद~~ पहले किया जाता है।

Integration testing :— Integration testing Software testing की एक विधि है जिसमें दो या दो से अधिक Software Units या Modules को जोड़कर एक किया जाता है तथा एक समूह में उनका परीक्षण किया जाता है। इस परीक्षण का मुख्य उद्देश्य integration units के मध्य bugs तथा faults को खोजना है तथा यूनिटों के मध्य कार्य, Performance तथा विश्वसनीयता की जाँच करना है। सामान्यतया Integration testing को एक Integration testing के द्वारा Perform testing से पहले किया जाता है। जब Unit testing कर लिया जाता है, तो प्रत्येक unit को एक-एक करके integrate किया जाता है और यह क्रिया तब तक चलती है जब तक कि सारी Unit को integrate नहीं कर लिया जाए।

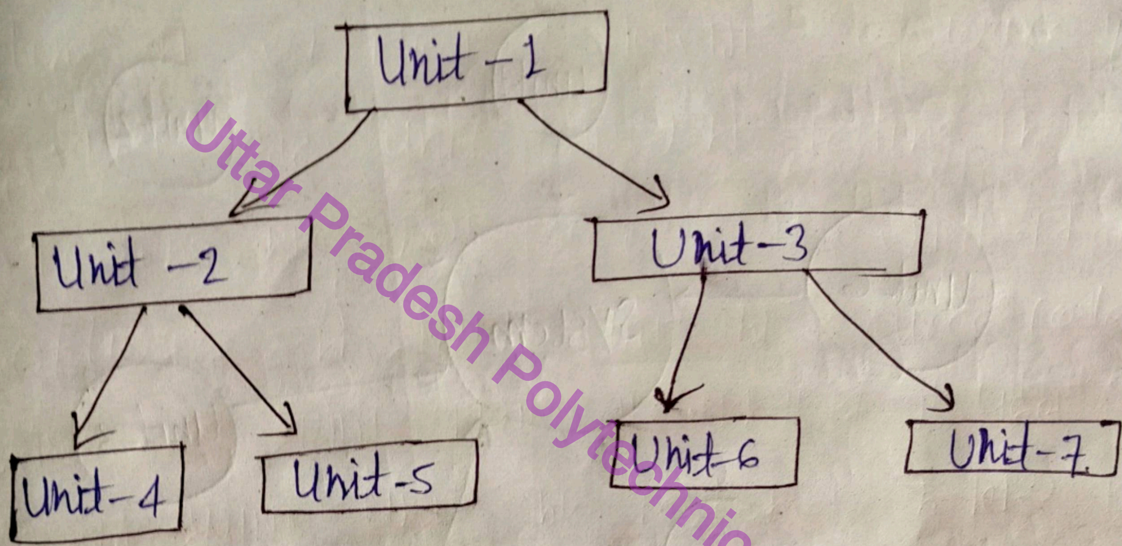


Integration testing

Uttam Prajesh Polytechnic (YouTube)

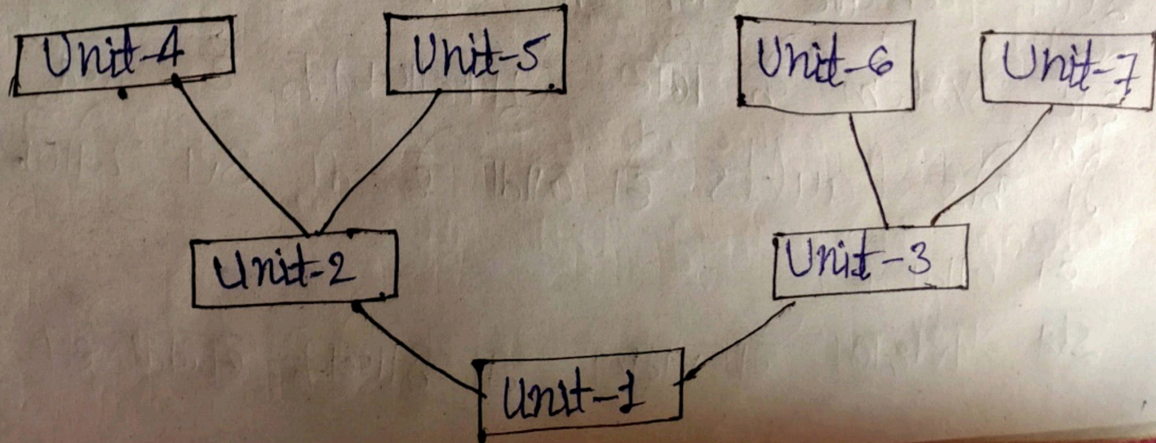
Integration testing के कई नियम है—

1. **Top-down Approach**:— इस integration testing में Top level के integratid units का सर्वप्रथम testing किया जाता है। तथा उसके बाद उसके नीचे वाले sub-units का परीक्षण किया जाता है।

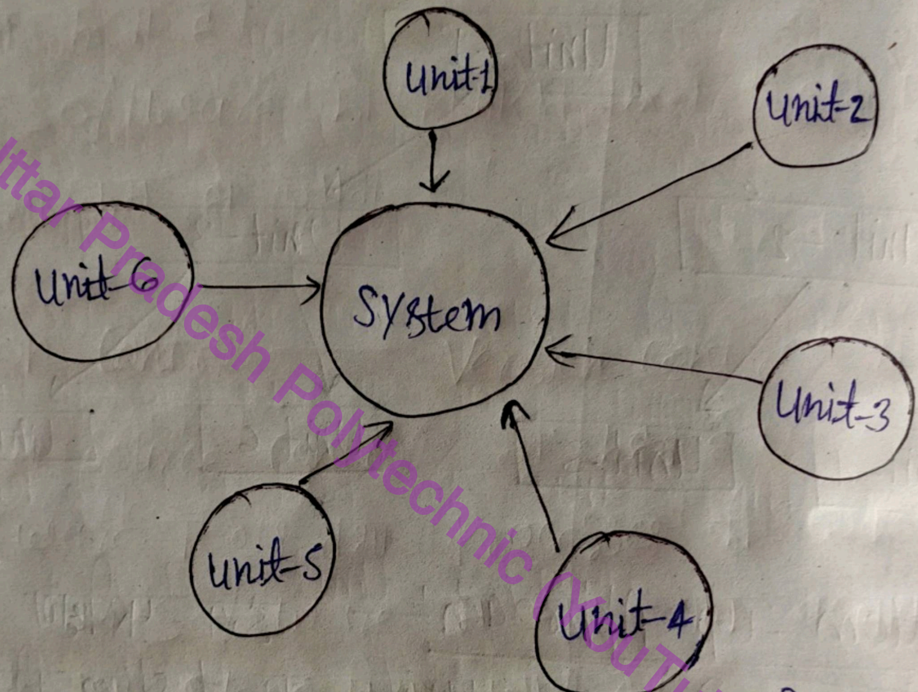


Uttar Pradesh Polytechnic (YouTube)

2. **Bottom-up Approach**:— इस परीक्षण में सबसे पहले bottom level के sub-units का testing किया जाता है। तथा उसके बाद ऊपर के मुख्य units को testing किया जाता है।



3. Big-Bang Approach :— इस प्रकार के testing में सभी यूनिटों को एक साथ integrate कर लिया जाता है, तथा इसके बाद एक समूह में सभी का testing कर लिया जाता है। इस integration testing में किसी एक unit को तब तक integrate नहीं किया जा सकता है। जब तक कि सभी units तैयार न हो जाएँ।



यह Testing बहुत ही risky होता है, क्योंकि अगर हमने सही प्रकार से documentation नहीं किया तो इसमें failure होने का खतरा बढ़ जाता है। यह Testing छोटे system के लिए सुविधाजनक है. परन्तु अगर system में कोई faults हो जाता है तो उसे खोजना मुश्किल हो जाता है।
इस परीक्षण में समय भी अधिक लगता है।

Uttar Pradesh Polytechnic (YouTube)

Software project management :-

Software Project Management (SPM) Software Project की योजना बनाने और उसे आगे बढ़ाने की उत्तम विधि है। यह Project management का एक भाग होता है जिसमें Software Project को plan, implement, monitor तथा control किया जाता है।

Uttar Pradesh Polytechnic (YouTube)

Software Project Management का वर्गीकरण :-

1. Conflict Management
2. Risk "
3. Requirement "
4. Change "
5. Software configuration management :- यह Software में change को trace करने तथा नियंत्रण करने की एक प्रक्रिया है। Software configuration management में revision control और baselines का उद्घाटन शामिल है।
6. Release management :

Advantages of Software Project :-

1. यह Software development को plan करने में मदद करता है।
2. इससे Software development को implement करना आसान हो जाता है।
3. यह Software को विकसित करने में लगने वाले समय को कम करता है।
4. इससे Software को विकसित करने में लगने वाली cost भी कम हो जाती है।

Uttar Pradesh Polytechnic (YouTube)

Version Control :- Version control को revision control या source control भी कहते हैं।

यह Software Tools तथा Processes की एक श्रेणी है जिसके द्वारा Software में होने वाले बदलावों (change) को track तथा store किया जाता है।

यह SCM (Software configuration management) का एक महत्वपूर्ण component है। अर्थात् "Version control एक management strategy है जिसके द्वारा किसी software, website या document में होने वाले change को manage किया जाता है"

वह System जो changes को track करने की सुविधा

प्रदान करता है, उसे version control system कहते हैं। यह system प्रत्येक बदलाव को एक विशेष प्रकार के database में store करता है।
इसमें यदि developers के द्वारा changes करने में कोई mistake हो जाती है तो software के पहले के version के साथ इसको compare किया जाता है और जो भी त्रुटि होती है उसे fix कर लिया जाता है।
version control का मुख्य उद्देश्य यह सुनिश्चित करना होता है कि changes planning के अनुसार घटित होनी चाहिए।

Uttar Pradesh Polytechnic (YouTube)

मुख्य version control system

1. GitHub
2. Gitlab
3. Beanstalk
4. Perforce
5. Apache Subversion
6. AWS Code commit
7. Microsoft team foundation server
8. Mercurial
9. CVS version control
10. Bitbucket

Types of Software Testing

- Functional Testing
  - Unit testing
  - Integration Testing
  - System testing
  - Interface Testing
  - Regression Testing
  - User Acceptance Testing
- Non-functional Testing
  - Security Testing
  - Documentation Testing
  - Installation Testing
  - Performance Testing
  - Reliability Testing

Uttar Pradesh Polytechnic (YouTube)

Feature of High level programming language

- मानव भाषाओं के समान समझने मे आसान है,
- Program के अनुकूल होती है, और Code करने मे आसान होती है,
- समझने मे आसान है,
- संकलन (Compile) & Debug मे आसान होती है

Translator

1. Interpreter — कोड → इंटरप्रेटर → O/P
2. Complier — Code → Complier → Machine Code → O/P
3. Assembler

3. Control Flow Based Design
(नियंत्रण प्रवाह आधारित डिजाइन)

High level language के आने के बाद Computer का उपयोग तेजी से बढ़ने लगा
Program ज्यादा जटिल और बड़े होने लगे जिससे उनका प्रबंधन मुश्किल होने लगा।
इसे आसान बनाने के लिए Control Flow Based Design तैयार किया गया,

Uttar Pradesh Polytechnic (YouTube)

Advantage of LOC

① लागत अनुमान में सबसे अधिक उपयोग किया जाने वाला मीट्रिक

② इस मीट्रिक की तुलना में इसके विकल्प में कई समस्याएँ हैं।

③ प्रयासों का आकलन इसका बहुत आसान है

Disadvantage of LOC

① कम कुशल कोड

② कोड की अधिक पंक्तियाँ लिखते हुए उत्पादकता बढ़ाता है।

③ यह जटिलता पर विचार नहीं करता है

Uttar Pradesh Polytechnic (YouTube)

(2) Decision trees & tables को विस्तारपूर्वक

Decision trees एक ऐसी संरचना है, जिसके द्वारा Data पर लागू होने वाली शर्तें तथा उन शर्तों के आधार पर लिए गये निर्णय को प्रदर्शित किया जाता है,

यह एक Data visualization का उत्तम तरीका है।

एक Decision Tree का प्रयोग निर्णय बनाने के लिए किया जाता है।

Decision Trees की संरचना एक Flow-चार्ट की तरह से होती है।

जिस प्रकार पेड़ में पत्तियाँ, जड़ तथा शाखाएँ होती उसी प्रकार इसमें leaf Node & class को प्रदर्शित करती है।

इसमें leaf node and Branches होते हैं, और सबसे ऊपर की Node को root कहते हैं।

Exampels → एक LED निर्माता company अपने ग्राहक के लिए डिस्काउन्ट-पॉलिसी को लागू करता है।

Type of customer | Size of order | Disc...

Discount policy for LED → Individual → 12 or more / Less than

manufacture → Shopkeeper or Retailer → 85 or more / 49-84 / 13-48 / Less than

IN... 50% 40% 30% 15%

Uttar Pradesh Polytechnic (YouTube)