

Index

unit-1:

1. Introduction, emergence of software, software development life cycle process modules.

unit-2:

=> Software Project management.

1) Project planning, Estimation.

=> software requirement specification.

1) Gathering, analysis, specification, characteristics, organization.

unit-3:

slw Design :

1. overview characteristic of good design,
2. function oriented slw design.
3. Object oriented slw design.
4. UML, Design patterns.

unit 4:

1. coding & unit testing - implementation.
2. coding standards & Guide lines.
3. code review, verification & Validation - unit testing.

unit-5:

Integration & System testing.

1. integration testing, system testing.

2. Black box testing.

3. white box testing.

4. Debugging techniques.

unit-6:

1. Software Quality.

2. SEI CMM

3. Software Reliability.

4. software safety

5. Risk analysis.

6. CASE.

7. iso 9001.

3/01/2019

Unit - 1

Software Engineering

Introduction:

Engineering := it is a application of science, tools and methods to find cost effective result to the problem is called as engineering.

Science
tools
methods

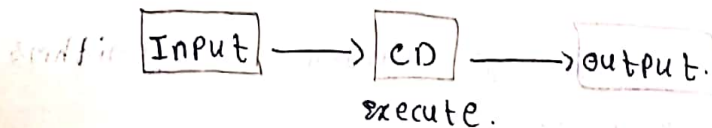
to produce cost effective result to the problem.

Software:

Instruction : set of rules.

Programme : set of instructions.

software : it is a set of programs executed in a computing devices it delivers given result by various functions.



Ex: mx player.

software engineering:

it is defined as systematic and disciplined and quantifiable approach to the development of software & maintain the software is called software engineering.

(or)

software engineering is defined as a establishment and use of application of the engineering principles. for obtaining reliable & feasible software that can run on the real time machine.

సాంఘిక మరియు గిరిజన సంక్షేమ శాఖల వారి ఉచిత సరఫరా

7. Artificial Intelligence software: Non-numerical algorithm to solve complex problems is called as Artificial intelligence software.

Ex: Robotics, recognizers (voice, face).

Applications of Software (or) Types of software:

* there is 7 types of application softwares.

1. System software:

A collection of programs written to service of other ^{user} programmers.

Ex: operating system, Linux, compilers, editors.

2. Application software:

it is also called as standalone software. it is mainly develop specific business needs. in the real time.

Ex: swiping machine, *Pos → point of sale.

3. Engineering/scientific software:

it is developed by large number of algorithms.

Ex: Astronomy, space, molecular biology.

4. Embedded software:

it control or manage hardware device.

Ex: ATM, mobiles, wms.

5. Product line software:

it is designed to a specific capability based by many different customers.

Ex: spreadsheet, word space, Antivirus.

6. web application software:

it is a client server computer problem. The client runs in a web browser.

Ex: Java, Perl.

7/01/2019
monday

Characteristics of software:

1. Security.
2. Accuracy → exact → correct.
3. Efficiency → Intime solution.
4. Useability

{

Operateability.

Learnability.

 understand.
5. Maintainability → modification.
6. Portability → moving one place to another.
7. Robustness → without weakness, constant work.
8. Integrity → prevented.

Emergence of software Engineering:

Early computer programming languages:

Emergence of software engineering can be divided into 6 phases

* software engineering technique is derived from many years as a result of collection of programs and series of innovations. There are mainly divided into six phases.

1. Early computer programming language.
2. High level programming language.
3. Control flow based Design.
4. Data structure oriented language.
5. Data flow oriented Design.
6. Object oriented Design.

సాంఘిక మరియు గృహీత సంక్షేమ శాఖల వారి ఉచిత సరఫరా

* Early computer programming languages:

1. it is very slow.
 2. Time: simple processing task take more time.
 3. Every programmer developed his own ^{individual} style of writing program.
 4. limit of code or size of code is 100 lines.
- Ex: Assembly level language.

II High level programming languages:

1. computers become faster with the introduction of semiconductor technology.
2. It reduced the time & effort required to develop software product.
3. limit of code is 1000 lines.

Ex: FORTRAN (Formula translation), COBOL (common business oriented language),
Algorithmic language (ALGOL).

* Both ECP and HLP languages are called as a Exploratory programming.

III. control flow Based Design:

1. size and complexity of program increasing, the exploratory programming language insufficient to solve the problem.
2. Experience programmers advised other programmers to attention design of the programs control flow structure.
3. The program control flow structure indicates sequence of instructions are executed. in order to help develop programs having good control flow structure, the flow-charting technique.

Ex: C, perl, modula, Pascal.

సాంఘిక నైపుణ్య గరిజన సంక్షేమ శాఖల వారి ఉచిత సరఫరా

IV. Datastructure oriented Design:

1. As computers more powerfull with introduced of integrated circuits they were used to solve more complicated software products.

2. Attention to the design of the important data structures of the program than to the design of its control structure.

Design techniques based on this principle are called data structure oriented Design Technique.

3. Using this technique first program's data structure designed.

4. Example of Datastructure oriented design technique is the TSP (Jackson Structured Programming). It is developed Michel Jackson (1975).

5. In TSP Program structure is derived from data structure representation.

V. Data flow oriented Design:

1. As computer became still faster and more powerful with the introduction of VLSI some new concepts, more complex and new products were needed to solve further challenging problems.

2. Therefore software engineers looked for more effective techniques for designing software products and soon data flow oriented technique were proposed.

3. This technique major items handled by a system must be first identified and then the processing required on these data items to produce the desired outputs should be determined.

4. The functions and data items that are exchanged b/w the different functions are represented in a diagram known as DFD.

Ex: engine store



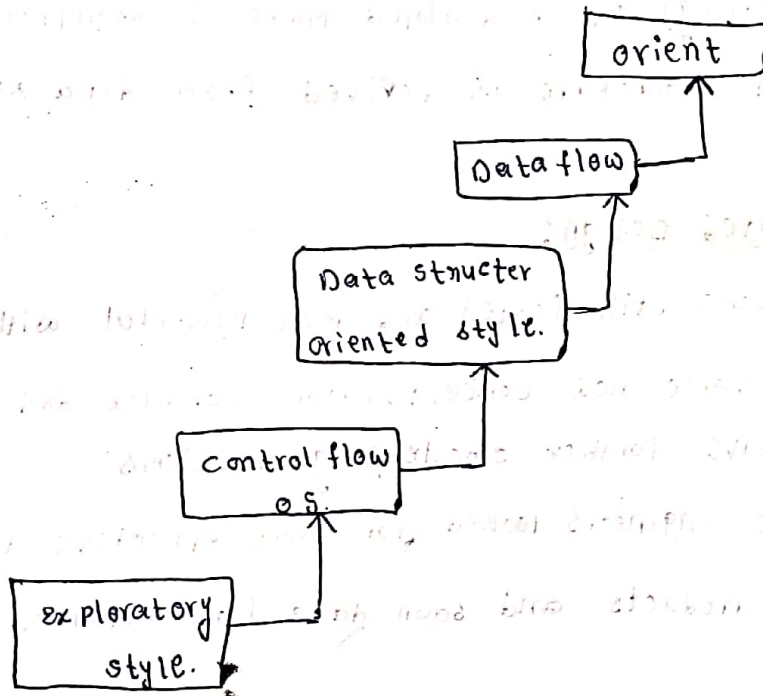
సాంఘిక మరియు గిరిజన సంక్షేమ శాఖల వారి ఉచిత సరఫరా

vi. Object oriented Design:

1. Data flow oriented Design Derived into object oriented design.
2. An OOD is appealing approach, where the natural objects (employee, stu) occurring in a problem first identified and then the relationships among the objects such as composition, reference, and inheritance are determined.
3. Each object essentially act as data hiding.
4. It is provide code and design reuse scope then offer, lower develop cost, more robust code, easien maintenance.

Other Developments:-

software design techniques evolved since the early days,



సాంఘిక మరియు గృహ సంక్షేమ శాఖల వారి ఉచిత సరఫరా

08/01/2019
Tuesday

Software Development Life Cycle :-

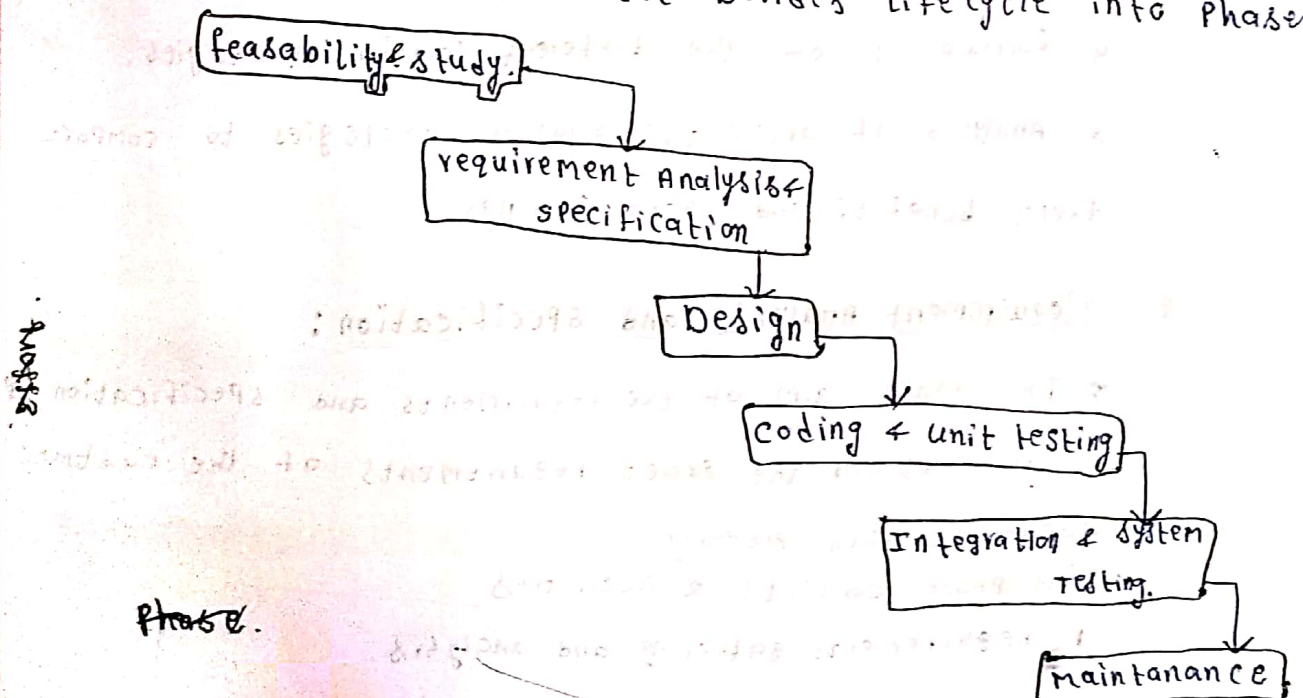
The software development life cycle categorized into 6 phases

1. Analysis.
2. Requirement gathering.
3. Design.
4. coding.
5. Testing.
6. Maintenance.

SDLC forms a common understanding of the activities among the software engineers and helps to develop a software in systematic and disciplined manner.

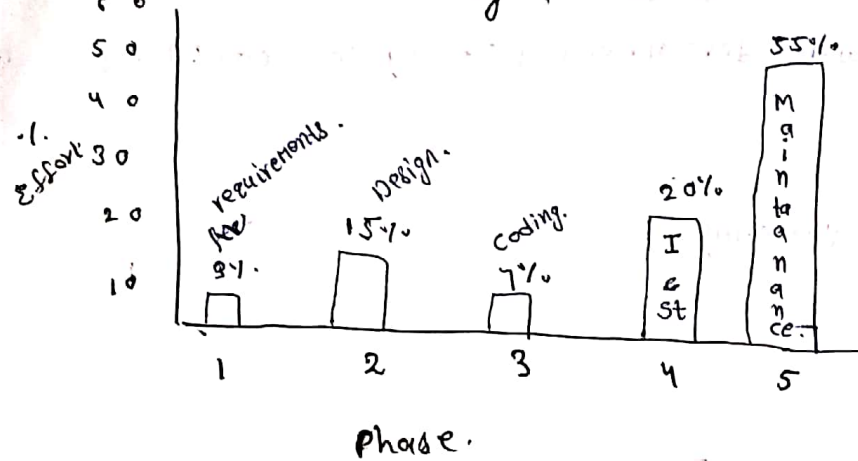
Classical waterfall model:

1. it is a unmistakable way to develop a SW.
2. This model is in theoretical way of a developing a software
3. Then why study this model?
Ans: Because of among all other life cycle models are essentially derived from classical waterfall model.
4. The classical waterfall model divides life cycle into phases.



సాంఘిక మరియు గౌరవనీయ సంక్షేమ శాఖల వారి ఉచిత సరఫరా

Effort Distribution among diff Phases.



We will put maximum effort for 55% of maintenance.

1. Feasibility Study :-

1. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product.
2. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product.
3. Abstract Problem: An abstract problem definition is rough description of the problem which considers only the important requirements and ignores the rest.
4. Formulation of the different solution strategies.
5. Analysis of alternative solution strategies to compare their benefits and shortcomings.

2. Requirement Analysis and Specification :

* The main aim of the requirements and specification phase is to understand the exact requirements of the customer and to document them properly.

This phase consists of 2 activities

1. Requirements gathering and analysis.

2. Requirement సాంఘిక మరియు గణిత సంక్షేమ శాఖల వారి ఉచిత సరఫరా specification.

Requirement gathering and analysis:

it is used to collect all relevant information from the customer regarding the product to be developed with a view to clearly understanding the customer requirements.

Requirement specification:-

The customer requirements identified during the requirements gathering and analysis activity are organized into a SRS.

SRS → software requirement specification.

* There is 2 important components.

1. functional requirements, non functional requirements.

⇒ functional requirements: A functional requirement is that it essentially specifies something the system should do.

Ex: add customer, print invoice, business rules, transaction correction.

⇒ Non-functional requirements: it identify the performance requirements.

Ex: Availability, Recoverability, maintainability.

3. Design:

The goal of the design phase is to transform the requirements specified in the SRS Document into a structure that is suitable for implementation in some programming languages.

⇒ The design phase the SW architecture is derived from the SRS document.

→ there are 2 types of Design approaches they are

1. traditional design approach.

2. object oriented design approach.

1. traditional Design approach:

=> The traditional Design approach is currently being used by many s/w development houses.

=> traditional Design consists of two different activities. first structured analysis of the requirements specification is carried out where the detailed structure of the problem is examined.

* During structured design the results of structure analysis are transformed into software Design.

* Dataflow Diagrams are used to perform structured analysis and to document the result.

2. object oriented design approach:

OOD is relatively new technique.

* In this technique various objects that occur in the problem domain and solution domain are first identified and the different relationships that exist among these objects are identified.

* The OOD approach has several benefit such as lower development time and lower development effort better maintainability of the product.

4. coding and unit testing:

* it is also called as implementation phase.

2. The purpose of the coding and unit testing phase of s/w development is to translate the software design into source code.

3. each component of the designs are implemented as a program module.

4. During this phase each module is unit testing tested to determine the correct working of all the individual modules.

5. Integration & system testing:-

1. integration of different modules is taken once they have been coded and unit tested.
2. During the integration and system testing phase the modules are integrated in a planned manner.
3. system testing is to ensure that the development system conforms to its requirements find out in the SRS document.
4. system testing usually consists of 3 different kinds of testing activities.

1. α -testing: it is the system testing performed by developers.

2. β -testing: it performed by friendly set of customers.

3. acceptancy-testing: it performed by the customers and giving the product delivery to determine whether to accept or reject the delivered product.

6. maintanance:-

it is a typical software product requires much more effort.

* a software product to its maintainance effort is roughly in the 10:60 ratio.

* maintainance mainly following 3 kinds of activities.

1. corrective maintainance: correcting the errors that were not discovered during the product development phase.

2. perfective maintainance: improving the implementation of the system and enhancing the functionalities of the system according to the customers requirements.

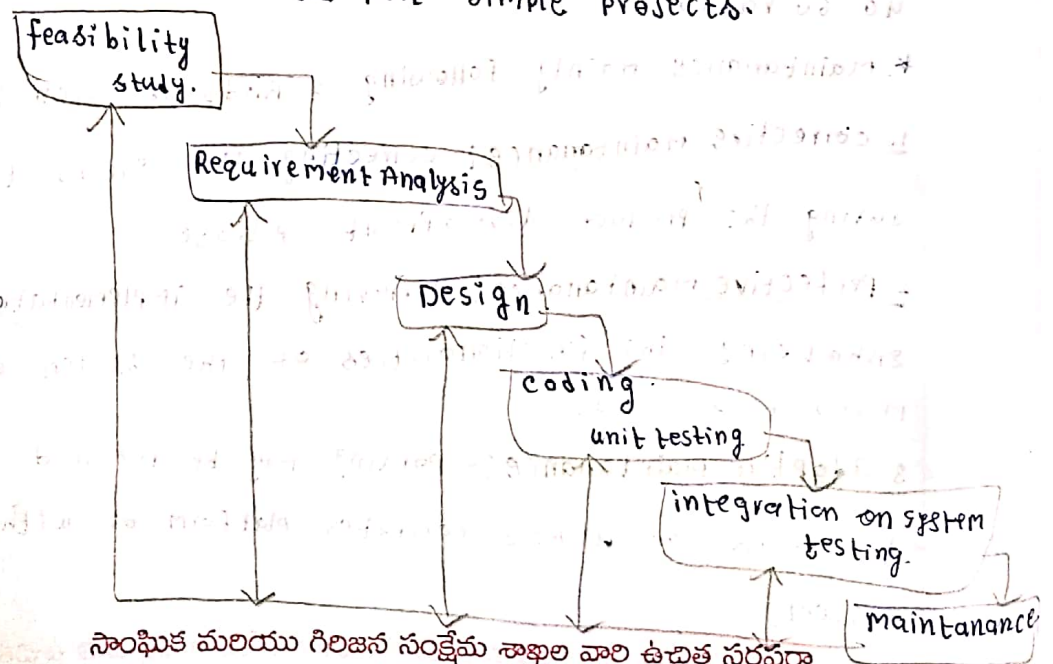
3. adaptive maintainance:- porting may be required to get the software to work on a new computer platform or with a new operating system.

10/01/2019.
Thursday.

Process models:-

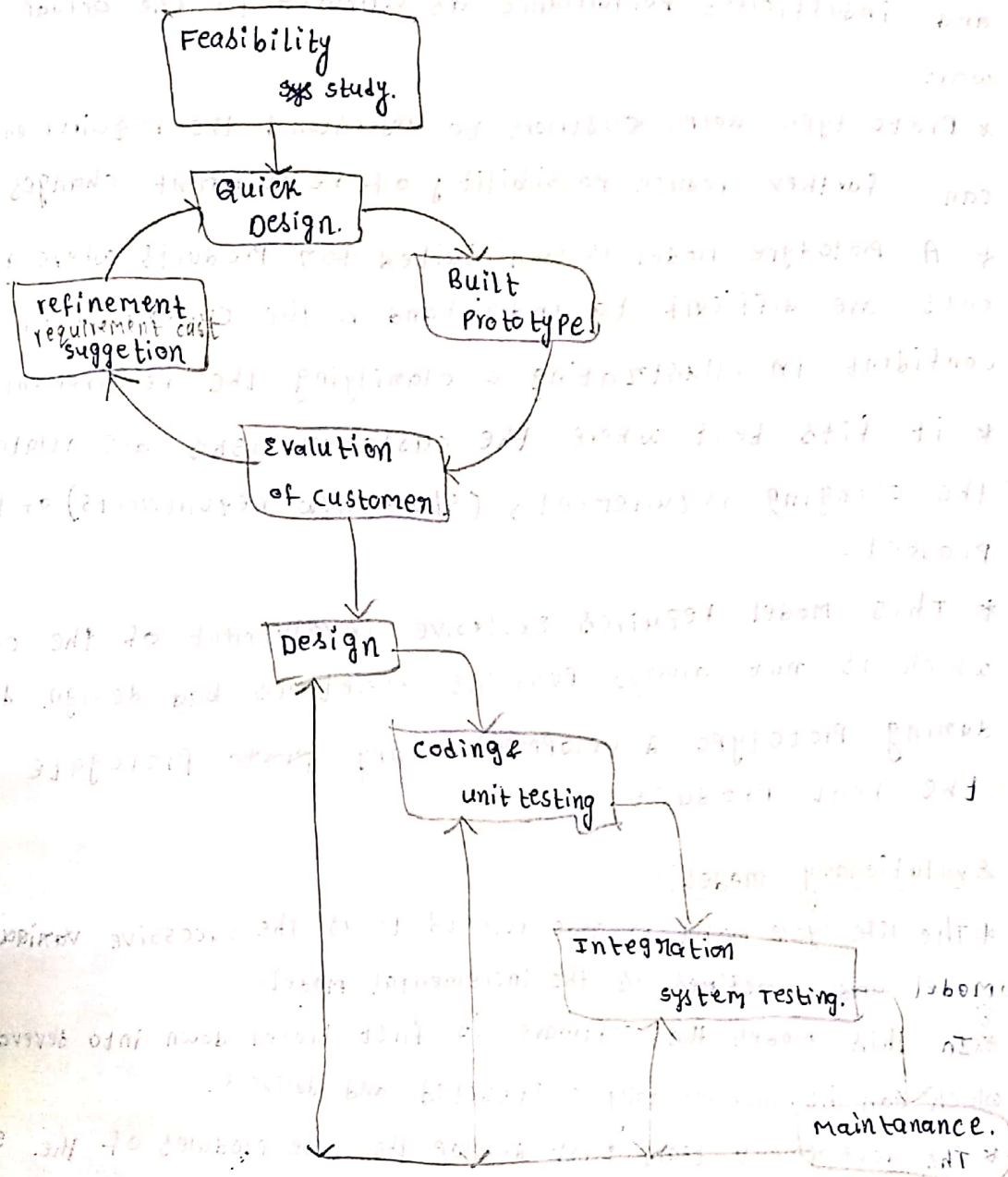
iterative waterfall model:-

1. The iterative waterfall model derived from classical waterfall model with backtracking at each Phase.
2. it is Proposed by R.W. Royce ⁱⁿ 1970.
3. the lifecycle organized similar to classical waterfall model Phases.
4. the only difference is backtracking of phases on detection of errors at any stage (or) any phase.
5. once error is detected again there is need to go back to the phase. where it was introduced.
6. on error detection at any phase, it may be required that the preceding and succeeding phases to be ~~phased~~ changed.
7. detecting and removing errors before software development process. is called phase containment error.
8. reduce the removing errors in early phases of a development reduce testing maintenance Effort.
9. Drawback : There is possibility of repeating phases which can slow down the efficiency.
10. This model is most suitable for simple projects.



సాంఘిక మరియు గిరిజన సంక్షేమ శాఖల వారి ఉచిత సరఫరా

Prototyping model :=



* Prototyping (partial working software) is a initially developed instead of developing the final product.

* IEEE defines the development in which emphasis is placed on developing prototypes early in the development processes is to permit early feedback & analysis.

* Support of the development process.

* Prototype development is a toy implementation which provides a chance to the customer to give feedback for final product development.

* A prototype provides limited functionalities, low reliability and insufficient performance as compared to the actual software.

* Prototype helps customer to understand the requirements that can further reduce possibility of requirement changes.

* A prototype model is well suited for products where requirements are difficult to understand & the customer is not confident in illustrating & clarifying the requirements.

* It fits best where the customer risks are related to the changing requirements (slw & h/w requirements) of the project.

* This model requires exclusive involvement of the customer which is not always possible sometimes bad design decision during prototype development may propagate to the real product.

24-01-2019
Thursday.

Evaluatory model:-

* The life cycle model is also referred to as the successive versions model and sometimes as the incremental model.

* In this model, the software is first broken down into several modules which can be incrementally constructed and delivered.

* The development team first develop the core modules of the system. The initial product skeleton is refined into increasing levels of capability by adding new functionalities in successive versions.

* Each evolutionary version may be developed using an iterative waterfall model in successive versions.

* Each evolutionary version may be developed using an iterative waterfall model of development.

* Each successive version of the product is fully functioning software capable of performing more useful work than the previous versions.

Rough Requirements specification.

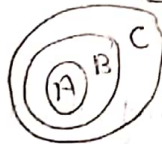
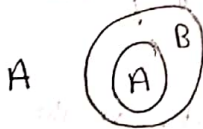
Identify the core and other parts to be developed immediately.

Develop the core part using an iterative waterfall model.

collect customers feedback and modify requirements.

Develop the next identified features using an iterative waterfall model.

All features complete, Evolutionary model.
maintenance.



A, B, C are modules of software product Evolutionary development of software product -

Advantages:

- * Here the user gets a chance to experiment with a partially developed software much before the complete version of the system is released.
- * The core modules get tested thoroughly, thereby reducing chances of errors in the core modules of the final product.

Disadvantages:

- * The successive versions mode is that for most practical problems it is difficult to divide the problems into several functional units which can be incrementally implemented and delivered.
- * it is used for very large products. it is easier to find modules for incremental implementation.

Spiral model:

The diagrammatic representation of this model appears like a spiral with many loops.

(1) Determine objects and identify alternate solutions.

(2) identify and resolve risks.

(3) Review and plan for the next phase.

(4) Develop the next level of the product.

సాంఘిక మరియు గిరిజన సంక్షేమ శాఖల వారి ఉచిత సర్వీస్

- * The exact number of loops in the spiral is not fixed.
- * Each loop of the spiral represents a phase of the software.
- * For example, the innermost loop might be concerned with feasibility study, next loop with requirement specifications the next one with design coding, testing, maintenance.
- * For instance, it is possible that for some project, the design might be accomplished over 3 or 4, consecutive loops and in some other project the design might be over in just one loop.
- * Each phase in the model is split into four sectors.
- * First section identifies the objectives of the phase and alternative solutions possible for the phase under consideration.
- * The second sector, the alternative solutions, are evaluated to select the best solution possible.
- * Third section consist of developing and verifying the next level of the product.
- * The fourth section concern receiving the results of the stages & traversed so far with the customer and planning the next iteration around the spiral.

Advantages:-

- * Requirement changes can be made at every stage.
- * Risks can be identified and rectified before they get problematic.

Disadvantages:-

- * It is based on customer communication if the communication is not proper then the software product that gets developed will not be up to the mark.