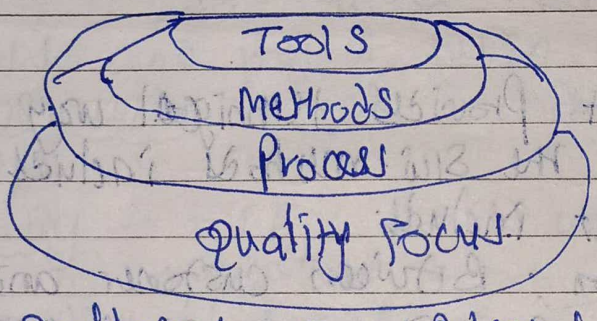
UNIT I

Software Engineering (SE)

SE is considered as a layered technology.

layers include:-

1. A quality Focus
2. Process
3. Methods
4. Tools.



Software Engg. layers.

I. Quality Focus:-

degree of goodness. Software can not be measured directly good quality s/w has following characteristics.

1. correctness :- correctness is degree to which s/w performs its required function.
2. maintainability - It is an ease with which s/w is maintained. maintenance of s/w implies change in s/w. if s/w is easy to maintain then quality of that s/w is good.
3. Integrity :- If Security is provided so that unauthorized user can not access data or information.

eg. integrity can be provided using password.

4. Usability :- The efforts required to use or operate the SW.

II Process :-

SW process defines a framework that includes different activities and tasks. In short process defines following what different activities and tasks are to be carried out during SW Engg. development. eg.

① What activities are to be carried out?
② What actions will be taken?
③ What tasks are to be carried out in a given action?

III Methods :-

It provides technical way stating how to implement the SW. method include collection of tasks that include

① Communication :- Between customer and developer.
② Requirement Analysis :- To start requirements in detail.
③ Analysis and design modelling :- To build model of SW to show requirements clearly.
④ Program construction :- Implementation of requirements using conventional programming language or automated tools.
⑤ Testing and support :- Test for errors and expected result.

IV] Tools :- SW Tools is nothing but automated support for SW development for eg.

① microsoft front page or microsoft publisher. can be used for web designing tool.

The Software process or process framework

A SW process is a collection of different activities. There are 5 generic process / frame work activities.

① Communication :-
The SW development process starts with communication between customer and developer. According to waterfall model customer must start all requirements at the beginning of project.

② planning :-
It includes complete estimation and scheduling and tracking.

③ Modeling :-
It includes detail requirement analysis and project design. (algorithm, flowchart)

④ construction :- It includes coding and testing steps:

① coding - design details are implemented using appropriate programming language.

② Testing - Testing is carried out.
① To check whether flow of coding is correct.
② To check out the errors of program.
③ To check whether program is giving expected output as per input specifications.

⑤ Deployment :-
It includes SW delivery, support and feedback from customer. if customer suggests some corrections or demands additional capabilities then changes are required for such corrections or enhancement.

Umbrella activities :- The umbrella activities occur throughout the process. They focus on project management tracking and control. The umbrella activities are:

1. S/w project tracking and control :- This is an activity in which s/w team can access progress and take corrective action to maintain schedule.
2. Risk management :- The risks that may affect project outcomes or quality can be analyzed.
3. S/w quality assurance :- There are activities required to maintain s/w quality.
4. Formal technical review :- It is required to access engineering work product to uncover and remove errors before they propagate to next activity.
5. S/w configuration management :- managing of configuration process when any change in the s/w occurs.
6. Work product preparation and production :- The activities to create model, documents, logs, forms and lists are carries out.
7. Reusability management :- It defines criteria for work product reuse.
8. Measurement :- In this technology activity, the process can be defined and collected. As to project and product measures are used to assist the s/w team in delivering the required s/w.

Process Model :- The process model can be defined as the abstract representation of process. The appropriate process model can be choosen based on standard abstract representation of process.

The S/W process model is also known as S/W development life cycle. (SDLC) model.

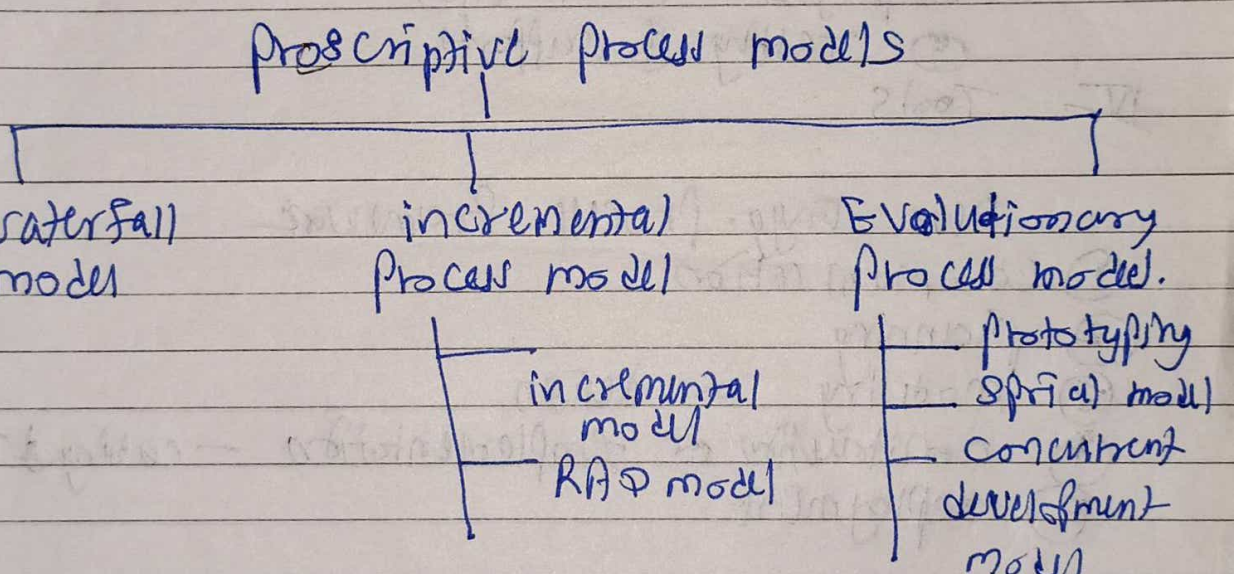


fig. Prescriptive process model.

* Need for process model.
1. S/W development can be done systematically.
2. Each team member will understand - what is the next activity and how to do it.
3. Each every process model consist of definate entry and exit criteria. for each phase.

SEMP
Page No.:
Date: / /

SW Engg. layers.

I Quality focus
1. Correctness
2. maintainability
3. integrity
4. usability

II Process — Activities & tasks

III Methods
1. communications
2. Requirement Analysis
3. Analysis & design modelling
4. program construction
5. Testing & support

IV Tools

SW Engg. process framework
1. communication
2. planning
3. modeling or design
4. construction or Implementation — coding & Testing
5. deployment

SEMP (3)

G. H. Raisoni College of Engineering & Management, Pune

Unit Test No.

Name of Student:

RAISONI GROUP, Roll No: Subject:

"a vision beyond" Class: Branch: Date: / /

Question No.	1	2	3	4	5	6	Total Marks	Signature of Examiner
Marks								

I Waterfall Model

The waterfall model is also called as linear sequential model or classic life cycle model. This model suggests a systematic, sequential approach to s/w development.

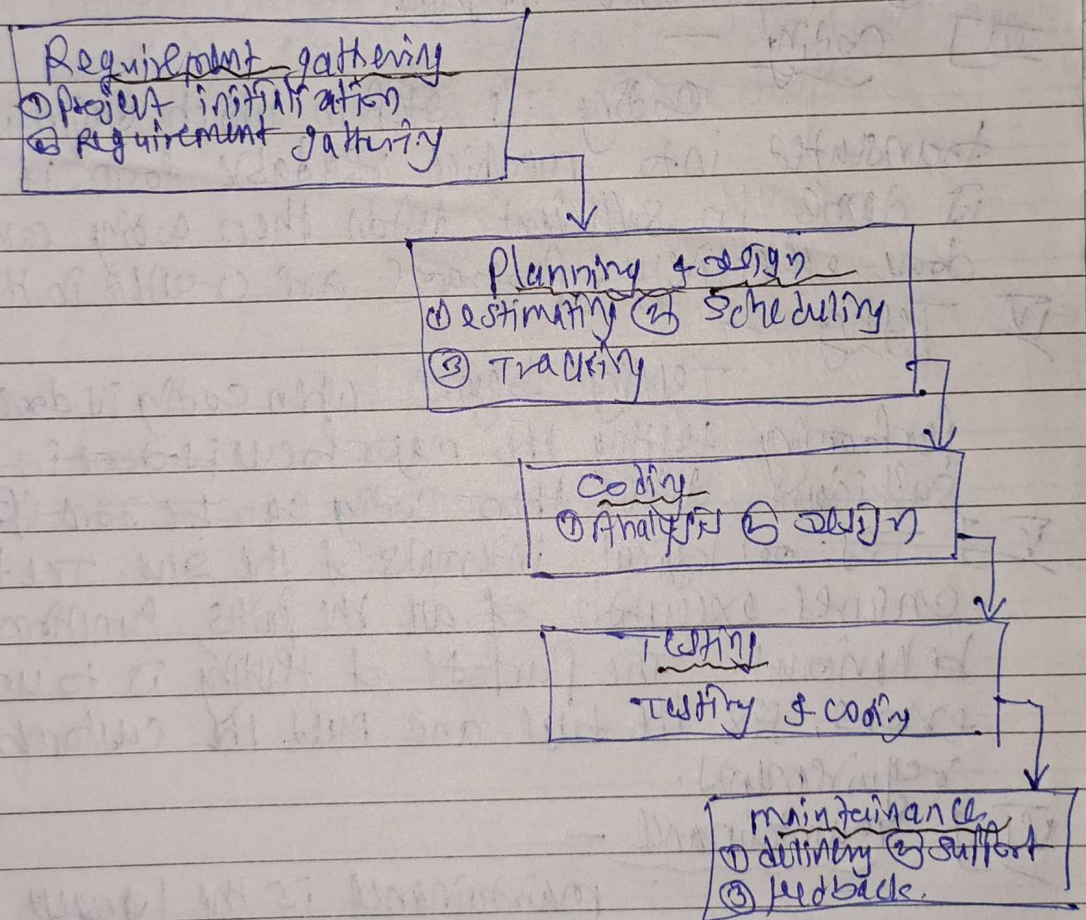


fig. Waterfall Model.

I] Requirement gathering and Analysis —

The basic requirements of the system must be understood by s/w engineer, who is also called Analyst. The information domain, function,

behavioural requirements of the system are understood. All these requirements are then well documented and discussed further with the customer for reviewing.

II] Design —

Design is intermediate step between requirements analysis and coding. Design focuses on program attributes such as.

1. Data structure
2. S/W Arch.
3. Interface representation
4. Algorithmic details

III] Coding —

Coding is step in which design is translated into machine readable form. if design is done in sufficient detail then coding can be done effectively. Programs are created in this phase.

IV Testing —

Testing begins when coding is done. While performing testing the major focus ~~is on~~ ~~Sufficient detail then coding can be done effectively~~ ~~V] Test~~ on logical internals of the S/W. The testing ensures execution of all the paths, functional behaviours. The purpose of testing is to uncover errors, fix the bugs and meet the customer requirements.

VI] Maintainance —

Maintainance is the longest life cycle phase. When the system is installed and put in practical use then error may get introduced. Correcting such errors and putting it in use is the major purpose of maintainance activity.

Advantages

1. Water fall model is simple to implement
2. for implementation of small systems waterfall is useful

→ disadvantage

① It is difficult to follow the sequential flow
② If some changes are made at some phase then it may cause some confusion.
③ The requirement analysis is done initially.
④ customers can see working model at end.
⑤ linear nature of waterfall model induces blocking states, because certain tasks may be dependent on some previous tasks

S/w — Collection of integrated programs
Engg. application of scientific and practical knowledge to invent design, build maintain & improve framework, process

Why SE required ?

1. To manage large S/w
2. for more scalability
3. cost management
4. manage the dynamic nature of S/w
5. better quality management.

Importance of SE

1. Reduce complexity
   minimize S/w cost
2. decrease time
3. Handling big project
4. Reliable S/w
5. effectiveness

Incremental Model in SDLC

Incremental model is a process of SW development where requirements are broken down into multiple standalone modules of SW development cycle. incremental development is done in steps from analysis, design, implementation, testing & maintainance.

1
1 +
1 +1
1 +1 =
1 +1 = 2

incremental model.

each iteration passes through the requirements, design, coding and testing phases. and each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.

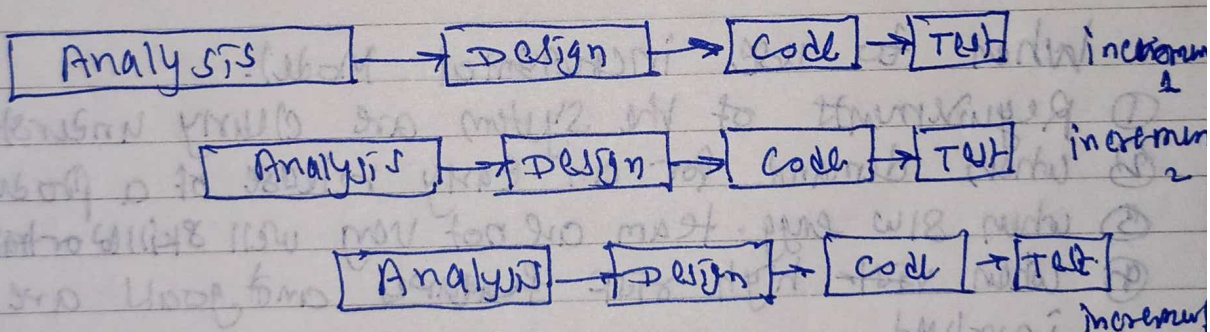


fig. increment model

The system is put into production when the first increment is delived. The first increment is often a core product where the basic requirements are addressed and supplementary features are added in the next increments. once the core product is analyzed by the the client, there is plan development for the next increment.

Characteristics of an incremental Module
① System development is broken down into many mini development projects.
② Partial Systems are successively built to produce a final total system.
③ Highest priority requirement is tackled first.
④ once the requirement is developed, requirement for that increment are Frozen

incremental Model phases.
1. Requirement Analysis — Requirement and specification of the SW are collected.
2. Design — some high end function are designed this stage.
3. Code :— coding of SW is done during this stage
4. Test :— once the system is deployed, it goes through the testing phase.

When to use incremental Model.
① Requirements of the system are clearly understood.
② when demand for an early release of a product arise
③ when SW engg. team are not very well skilled or trained
④ when high risk features and goal are involved.
⑤ methodology is more in use for web application and product based companies.

Advantage :— SW will be generated quickly during the SW life cycle.
⇒ Is advantage :— It requires a good planning designing.

Advantages

1. SIW will be generated quickly during the SIW life cycle.
2. It is flexible and less expensive to change requirements and scope
3. Throughput the development stages changes can be done.
4. This model is costly compared to others
5. A Customer can respond to each building.
6. Errors are easy to be identified.

Disadvantages

① It requires a good planning designing.
② problems might cause due to system architecture as such not all requirements collected up front for the entire SIW life cycle
③ each iteration phase is rigid and does not overlap each other.
④

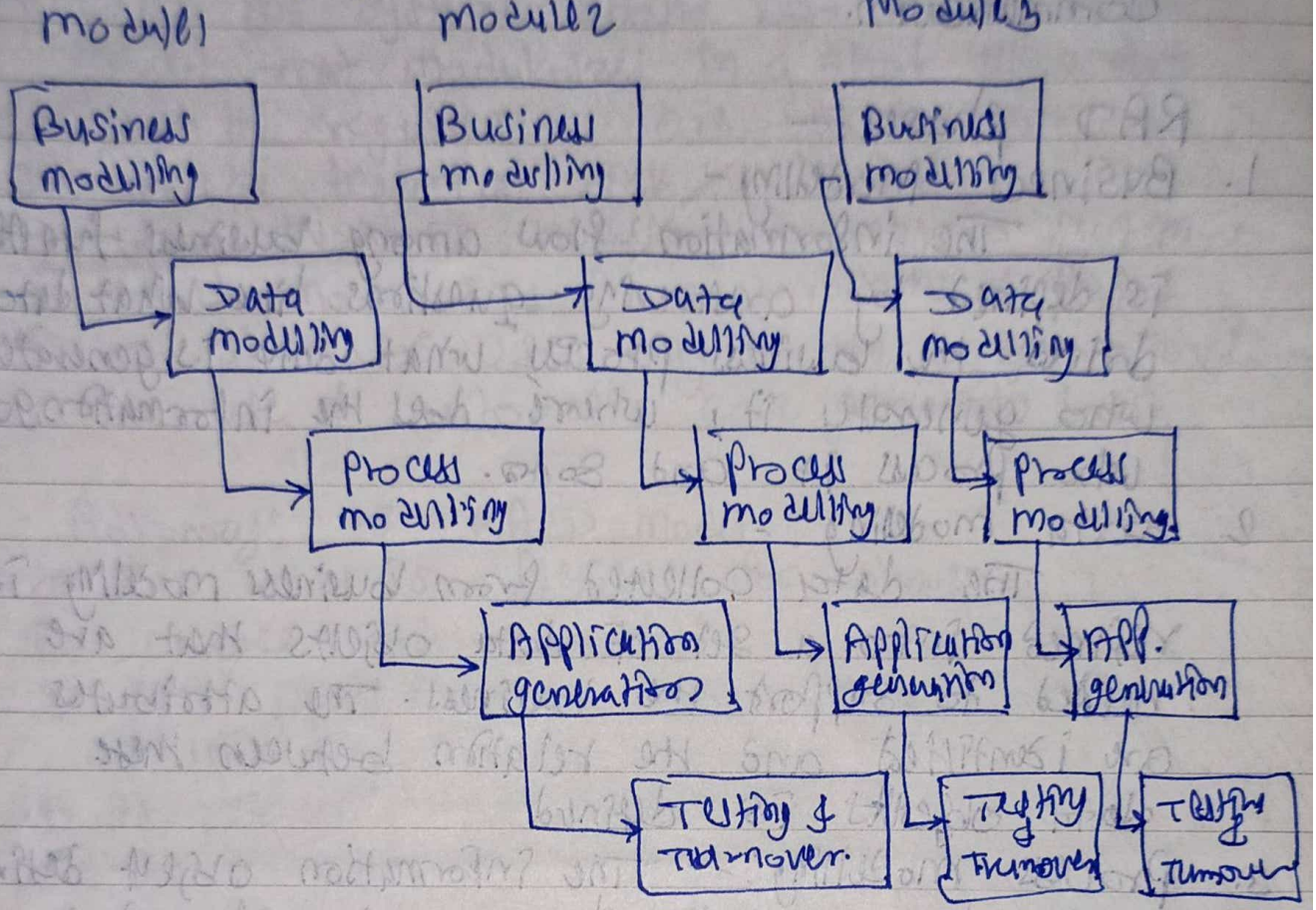
Waterfall Model

① Need of detailed doc. in waterfall model necessary.
② early stage planning is necessary
③ High risk
④ long waiting time for running s/w
⑤ can not handle large project.
⑥ changes are difficult.
⑦ cost is low
⑧

incremental Model

① doc. needed but not too much.
② early stage planning is necessary.
③ low risk
④ short waiting time in running s/w.
⑤ cannot handle large project.
⑥ changes are easy.
⑦ cost is low.

RAD (Rapid Application development) Model



RAD is a linear sequential s/w development process model that emphasizes a concise development cycle using element based construction approach. If the requirements are well understood and described, and the project scope is constraint, the RAD process enables a development team to create a fully functional system within a concise time period.

RAD is a concept that products can be developed faster and of higher quality through:

① gathering requirements using workshops or focus group.
② prototyping and early, reiterative user testing of designs.
③ The re-use of s/w components.
④ A rigidly paced schedule that refers design improvements to the next product version.

⑤ less formality in reviews and other team communication.

RAD phases:-

1. Business Modelling -
   The information flow among business functions is defined by answering questions like what data drives the business process, what data is generated, who generates it, where does the information go, who process it and so on.
2. Data modelling :-
   The data collected from business modelling is refined into a set of data objects that are needed to support the business. The attributes are identified and the relation between these data objects is defined.
3. Process modelling :- The information object defined in the data modelling phase are transformed to achieve the data flow necessary to implement a business function. Processing descriptions are created for adding, modifying, deleting or retrieving a data object.
4. Application generation :-
   Automated tools are used to Facilitate construction of the SW, even they use the 4th GL Technology.
5. Testing & Turnover - many of the programming components have already been tested since RAD emphasis reuse. This reduces the overall testing time. But the new part must be tested and all interfaces must be fully exercised.

When to use RAD model

① When the system should need to create the project that modularized in a short span time.
② When the requirements are well known
③ When the technical risk is limited
④ When there is necessary to build a system, which modularised in 2-3 months of period.
⑤ It should be used only if the budget allows the use of automatic code generation tools.

Advantage of RAD model.

① This model is flexible for change.
② changes are adoptable.
③ each phase in RAD brings highest priority functionality to the customer.
④ it reduced development time.
⑤ It increases the reusability of features

⇒ Disadvantage of RAD model.

① It required highly skilled designers
② All application is not compatible with RAD
③ For smaller projects, we cannot use the RAD model.
④ on the high technical risk, it is not suitable
⑤ Required user involvement

Difference between RAD model and incremental model.

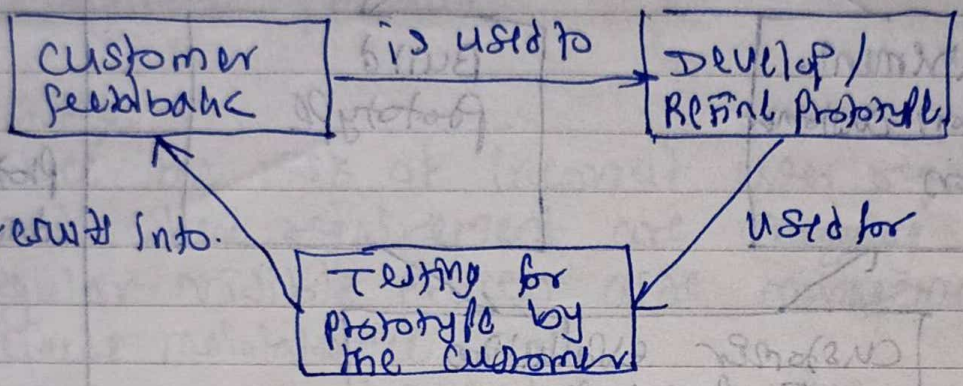
RAD model	incremental model.
① components of functions are developed in parallel	① product is analyzed, designed, implemented.
② early stage planning is not necessary.	② early stage planning is necessary.
③ used for medium or small project.	③ used for large project.
④ low risk	④ low risk
⑤ small team size	⑤ large team size
⑥ easily changes can be done.	⑥ difficult to do changes.
⑦ Testing is done after coding.	⑦ testing is done after every iteration of phase.
⑧ return to previous stage/ phase is possible.	⑧ Returning to previous stage/ phase is possible.
⑨ cost is low	⑨ cost is high.

Difference between RAD and water fall model.

~~waterfall~~ RAD model	RAD ~~water fall~~ model
① classical model	① Rapid application development
② High risk	② low risk.
③ large team size	③ small team size
④ used for small project.	④ used for medium or small project.
⑤ changes can be done at starting only	⑤ changes can be made anytime.
⑥ product delivered after completion. Stages.	
⑦ long waiting time.	⑦ short waiting time

Prototyping model

Prototyping is defined as the process of developing a working replication of a product or system that has to be engineered. it offers a small scale of the end product and is used for obtaining customer feedback as follows:



The prototyping model is one of the most popularly used SDLC development life cycle model. This model is used when the customers do not know the exact project requirements beforehand. In this model as per customer feedback repeatedly till a final acceptable prototype is achived which forms the basis for developing the final product.

The prototype model requires that before carring out the development of actual SDLC, a working prototype of the system should be built. A prototype is a toy implementation of the system. A prototype usually turns out to be a very crude version of the actual system, possible exhibiting limited functional capabilities, low reliability and inefficient performance as compared to actual SDLC. in many instances, the client only has a general view of what is expected from the SDLC product. In such a scenario where there is an absence of detailed information regarding the I/P to the system.

The processing needs, and the of requirement, the prototyping model may be employed.

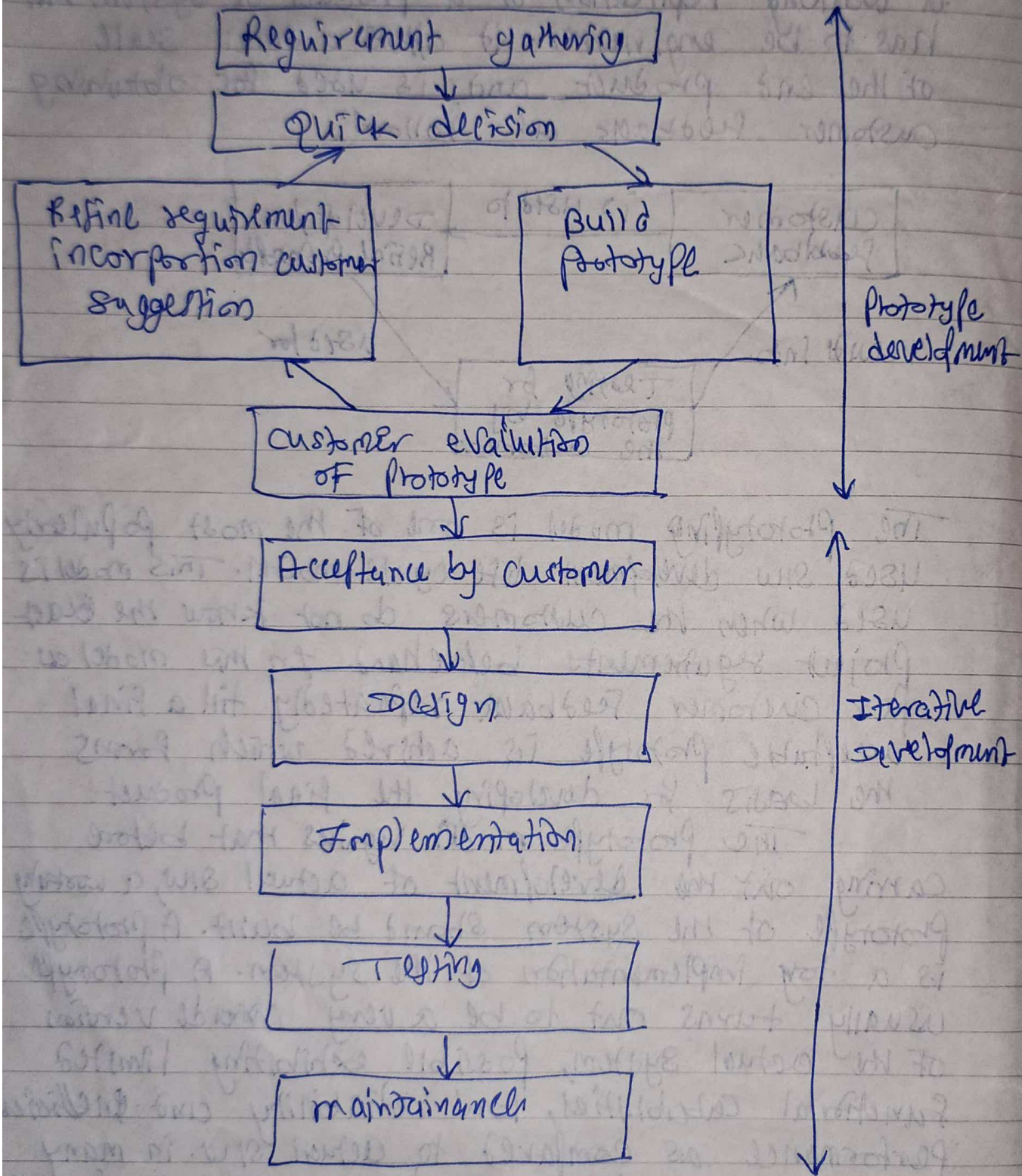


Fig. prototype model

Steps of Prototype model

① Requirement gathering & Analyst
② quick decision
③ Build a prototype
④ Assessment or user evaluation
⑤ Prototype refinement
⑥ Engineer product.

Advantage

1. Reduce the risk of incorrect user requirement.
2. Good where requirement are changing
3. Regular visible process aids management.
4. Reduce maintainance cost.
5. Errors can be detected much earlier as the system is made side by side.

Disadvantage

1. An unstable/ badly implemented prototype often becomes the final product.
2. Require extensive customer collaboration.
   - Costs customer money
   - needs committed customer
   - difficult to finish if customer with draw.
   - may be too customer specific, no broad market.
3. difficult to know how long the project will last
4. easy to fall back into the code and fix without proper requirement analysis, design, customer evaluation and feedback.
5. prototyping tools are expensive
6. Special tools & techniques are required to build a prototype
7. It is a time consuming process.

When to use prototype model

① When the desired system needs to have a lot of interaction with the end users.

② online system, web interfaces have a high amount of interaction with end users.

Difference between waterfall model and prototype model.

Waterfall model	Prototype model
① Works in sequential method.	① Prototype is built, tested and then refined as per customer needs.
② It give emphasis on risk analysis	② It does not give emphasis on risk analysis.
③ High risk low risk project	③ Suitable for high risk project
④ quick initial reviews are possible.	④ quick initial reviews are not possible.
⑤ It is best suited when the customer requirements are clear.	⑤ It is best when the requirements of the client is not clear and supposed to changed.
⑥ User involvement is only at the beginning.	⑥ User involvement is start to end.
⑦ It supports automatic code generation as results in minimal code writing.	⑦ It does not support automatic code generation.
⑧ The complexity of an error increases as the nature of the model each phase is sequential of the other.	⑧ complexity of error is low.
⑨ difficult to do changes.	⑨ changes can be done easily.

Difference between prototype model and incremental model.

Prototype model	incremental model.
① prototype is built, tested and then refine as per customer needs.	① where the product is analyzed, designed, implemented and tested incrementally.
② suitable for high risk projects.	② can handle large project.
③ cost is low	③ cost is low
④ Flexibility to change in prototype model is easy.	④ Flexibility to change in incremental model
⑤ It does not support automatic code generation	⑤ It supports automatic code generation as results in minimal code writing.
⑥ It does not give emphasis on risk analysis	⑥ It give emphasis on risk analysis.
⑦ User involvement is high.	⑦ User involvement is only at the beginning

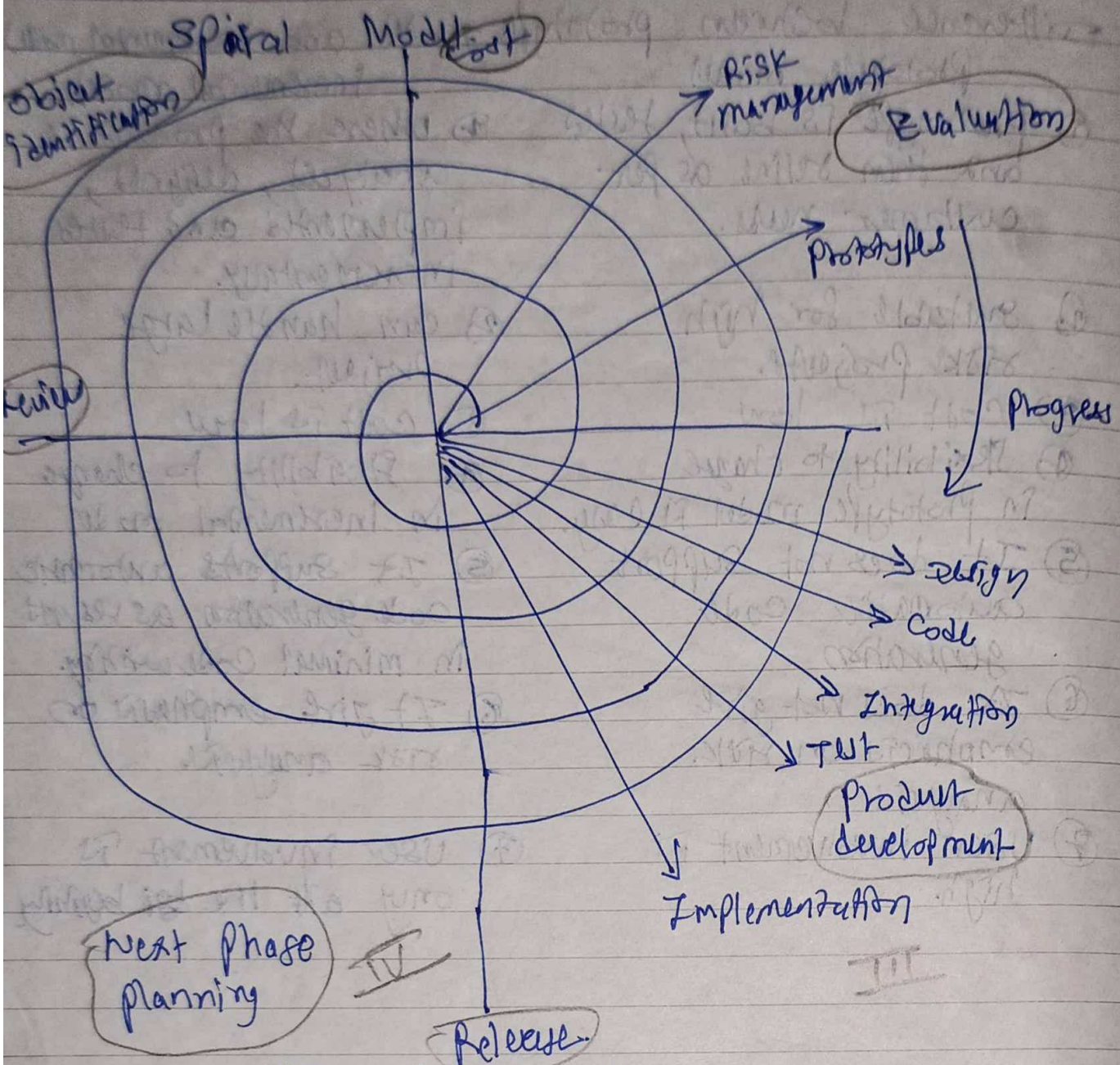


fig. Spiral Model

Each cycle in the spiral is divided into 4 parts.

1] Objective setting/ identification :-
each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets and the constraints that exists.

2] Risk Assessment and reduction — The next phase in the cycle is to calculate these various alternatives based on the goals and constraints.

The focus of evaluation in this stage is located on the risk perception for the project.

3] Development and validation :- The next phase is to develop strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.

4] Planning :- Finally, the next step is planned. The project is reviewed and a check made whether to continue with a further period of the Spiral. If it is determined to keep, plans are drawn up for the next step of the project.

The development phase depends on the remaining risks. For example, if performance or user interface risks are treated more essential than the program development risks, the next phase may be an evolutionary development that includes developing a more detailed prototype for solving the risks.

The risk driven feature of the Spiral model allows it to accommodate any mixture of a specification-oriented, prototyped-oriented, simulation-oriented, or another type of approach. An essential element of the model is that each period of the Spiral is completed by a review that includes all the products developed during that cycle, including plans for the next cycle.

When to use Spiral model

① when deliverance is required to be frequent
② when the project is large.
③ when requirements are unclear and complex
④ when changes may require at any time.
⑤ large and high budget projects.

Advantage:-
1. High amount of risk analysis
2. useful for large and mission critical projects

Disadvantages:-
1. can be a costly model to use
2. Risk analysis needed highly particular expertise.
3. Doesn't work well for smaller projects.

Difference between waterfall model and spiral model

Waterfall model	Spiral model
① Simple & easy	① more complex.
② work in sequential method.	② work in evaluationary method.
③ errors or risks are identified and rectified after the completion of stage.	③ errors or risks are identified and rectified earlier.
④ adopted by customer.	④ adopted by developers.
⑤ applicable for small project.	⑤ used for large project.
⑥ requirements & early stage planning is necessary.	⑥ requirements & early stage planning is necessary if required.
⑦ difficult to do changes	⑦ changes are easily changed. / accepted
⑧ high risk.	⑧ low risk
⑨ less expensive	⑨ expensive
⑩ customer involvement is minimum.	⑩ customer involvement is more or high.

Difference between waterfall model and Spiral model.

waterfall model	Spiral model.
① simple and easy	① more complex.
② works on sequential method.	② works on evaluatory method.
③ errors or risks are identified and rectified after the completion of stages.	③ errors or risks are identified and rectified earlier.
④ model adapted by customers.	④ adopted by developers.
⑤ applicable for small projects.	⑤ used for large project.
⑥ requirements and early stage planning is necessary.	⑥ requirements and early stage planning is necessary if required.
⑦ changes are difficult	⑦ changes are easy
⑧ high risk	⑧ low risk
⑨ less cost	⑨ expensive.
⑩ customer involvement is minimum.	⑩ customer involvement is maximum.

Difference between prototype and spiral model.

Prototype model	Spiral model
① prototype is built, tested and then refined as per customer needs.	① risk driven sw development model is made with features of incremental, waterfall or evolutionary prototyping model.
② referred to as rapid or closed ended prototype	② refered to as meta model.
③ does not give emphasis on risk analysis	③ It takes special care about risk analysis and alternative solution is undertaken.
④ customer interaction is continous untill the final prototype is approved.	④ there is no continous customer interaction
⑤ It is best suited when the requirement of the client is not clear and supposed to be change.	⑤ best suited when the customer requirements are clear.
⑥ cost effective quality improvement is very much possible.	⑥ cost effective quality improvement is not possible.
⑦ improvement of quality dold not increases the cost of product.	⑦ improvement of quality can increase the cost of product.
⑧ large scale project is maintained	⑧ low to medium project size is maintained.

Date :
Page :

Difference between spiral model and incremental model:

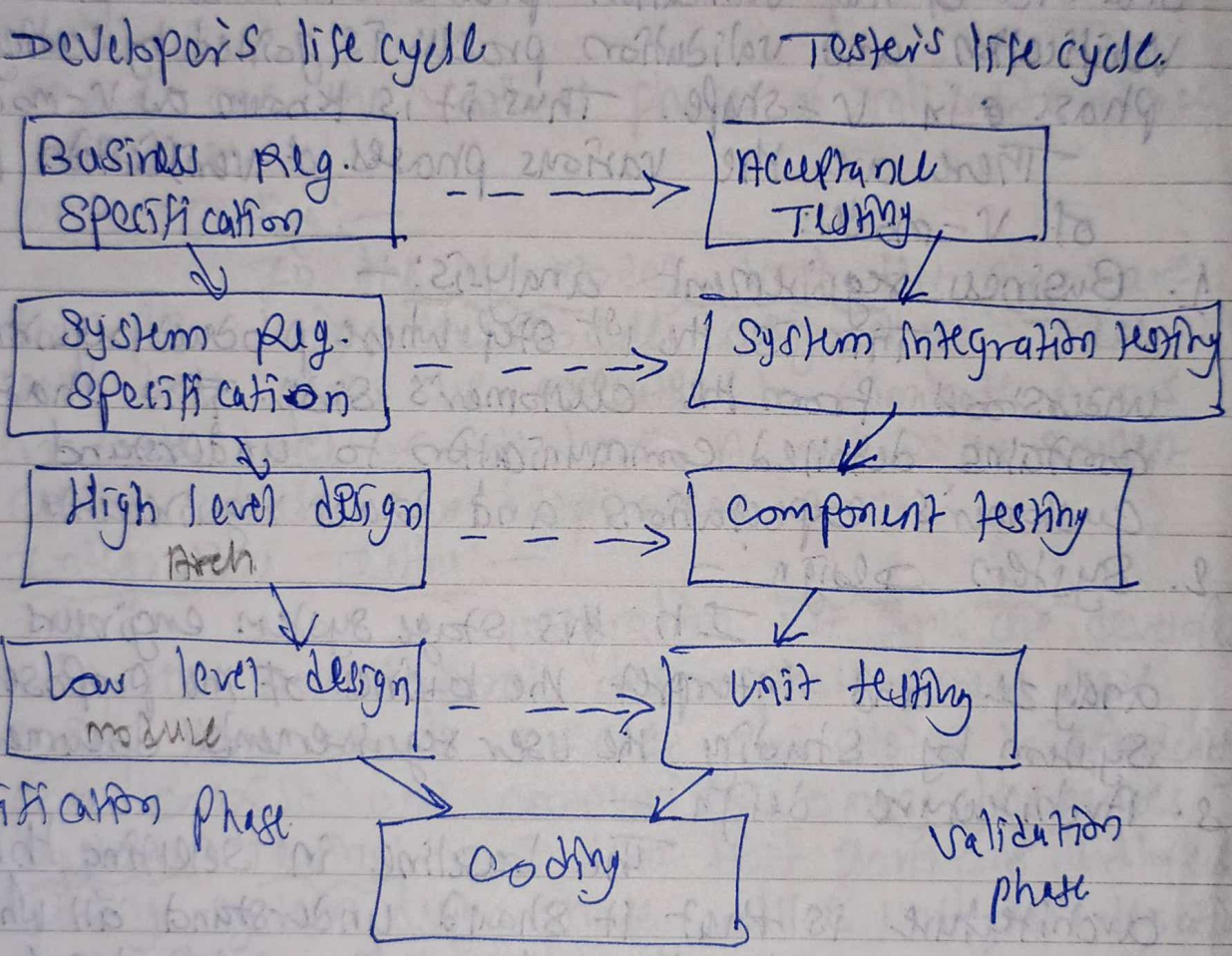
Spiral model	incremental model
① Features of incremental, waterfall or evoluationary prototyping model.	① product is analyzed, designed, implemented and tested incrementally.
② early stage planning	② early stage planning.
③ To do changes in model is difficult.	③ to do changes in model is easy.
④ low risk.	④ low risk
⑤ costly	⑤ low cost
⑥ handle large project	⑥ can not handle large project.
⑦ overlapping of phases are not possible.	⑦ overlapping of phases is possible
⑧ Testing is done is spiral model at the end of the eng. phase.	⑧ Testing is done in incremental model every iteration of phase.
⑨ Returning to previous stage/ phase in spiral model is possible.	— 11 —
⑩ large team required.	⑩ large team is required.

Difference between RAD model and Spiral model

RAD model	Spiral model
① Components or functions are developed in parallel as if they were mini projects	② model is made with features of incremental, waterfall or evolutionary prototyping models.
② requirements and early stage planning is not necessary.	② requirements and early stage planning is required
③ used in between large & small project.	③ used for large project.
④ low amount risk.	④ high amount risk.
⑤ small team size is required	⑤ large team is required.
⑥ Flexibility to easily change	⑥ Flexibility to difficult to change.
⑦ overlapping of phases is possible.	⑦ overlapping of phases is not possible.
⑧ Testing is done in rad model after completion of coding.	⑧ Testing is done in spiral model at the end of the engg. phase.
⑨ low cost	⑨ expensive

V- Model

V- model is also referred to as the verification and validation model. In this, each phase of SDLC must complete before the next phase starts. It follows a sequential design process same as the waterfall model. Testing of the device is planned in parallel with a corresponding stage of development.



Verification:- It involves a static analysis method done without executing code. It is the process of evaluation of the product development process to find whether specified requirements met.

Validation :- It involves dynamic analysis method functional, non functional testing is done by executing code. Validation is the process to classify the s/w after the completion of the development process to determine whether the s/w meets the customer expectations are requirements.

So V-model contains verification phases on one side of the validation phases on the other side. Verification and validation process is joined by coding phase in V-shape. Thus it is known as V-model.

There are the various phases of verification phase of V-model:

1. Business requirement analysis:-
This is the 1st step where product requirements understood from the customer's side. This phase contains detailed communication to understand customer's expectations and exact requirements.
2. System design:-
In this stage system engineers analyze and interpret the business of the proposed system by studying the user requirements document.
3. Architecture design:-
The baseline in selecting the architecture is that it should understand all which typically consists of the list of modules brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology detail etc. The integration testing model is carried out in a particular phase.
4. module design:-
In this module design phase, the system breaks down into small modules. The detailed design of the modules is specified, which is known as low level design.

5. Coding phase:-

After designing, the coding phase is started. Based on the requirements, a suitable programming language is decided. There are some guidelines and standards for coding. before checking in the repository, the final build is optimized for better performance, and the code goes through many code reviews to check the performance.

There are the various phases of validation phase of V-Model.

① Unit Testing:-

In the V-model, unit test plans are developed during the module design phase. These unit test plans are executed to eliminate errors at code level or unit level. A unit is the smallest entity which can independent exist.

② Integration Testing:-

integration test plans are developed during the architectural design phase. These tests verify that groups created and tested independently can coexist and communicate among themselves.

③ System Testing:- System test plans are developed during system design phase. unlike unit and integration test plans, system tests plans are composed by the client. business team. System test ensures that expectations from an application developer are met.

④ Acceptance Testing:-

Acceptance testing is related to the business requirement analysis part. It includes testing the SW product in user atmosphere. Acceptance test reveal the compatibility problems with different systems.

which is available within the user atmosphere. It conjointly discovers the non functional problems like load and performance defects within the real user atmosphere.

When to use V-model:

1. When the requirement is well defined and not ambiguous
2. The V-shaped model should be used for small to medium size project where requirements are clearly defined and fixed.
3. The V-shaped model should be chosen when sample technical resources are available with essential technical expertise.

Advantage:-

1. Easy to understand.
2. Testing methods like planning, test designing happens well before coding.
3. Save lot of time.
4. Avoid the downward flow of the defects
5. Works well for small plans where requirements are easily understood.

Disadvantage:

1. Very rigid and least flexible
2. Not good for a complex project
3. Slw is developed during the implementation stages, so no early prototype of the slw are produced.
4. If any change happen in the midway, then the test documents along with the required documents, has to be updated.

Difference between V- model and water fall model

Water fall model	V - model
① cost is low	① expensive
② model is simple	② model intermediate.
③ Not flexibility	③ flexible
④ sequential execution process.	④ sequential
⑤ steps are move in a linear way	⑤ don't move in linear way.
⑥ Re-usability is limited	⑥ Reuse for some extent
⑦ User involvement at beginning	⑦ User involvement at beginning.
⑧ testing activities start after the development activities are over.	⑧ testing activities start with the 1st stage
⑨ No. of defects are less.	⑨ as comparision more defect.
⑩ Requirement specification is necessary in beginning	⑩ —11—
⑪ less used now a days	⑪ widely used.

Difference between V- model and incremental model.

V- model	incremental model.
① SLC development and testing are not concurrent	① product is analyzed, designed, implemented and tested incrementally.
② testing activities start with the 1st stage	② Testing is done in incremental model after every iteration of phase.
③ little Flexible.	③ more flexible
④ expensive	⑥ less costly
⑤ good quality product	⑤ low quality product
⑥ not iterative	⑥ iterative

Difference between V- model and RAD model.

V- Model	RAD model.
① development and testing are not concurrent	① components or functions are developed in parallel.
② User involvement at beginning	② —11—
③ little flexible	③ more flexible
④ expensive	④ less cost
⑤ testing activities start with the 1st stage	⑤ Testing is done after coding.
⑥ quality is high	⑥ low quality
⑦ not iterative	⑦ iterative.

Difference between V-model and spiral model

V-model	Spiral model
① development and testing are not concurrent	① development and testing are concurrent.
② testing activities start with the 1st stage	② testing is done at end.
③ costly	③ less cost
④ little flexible	④ difficult
⑤ product is high quality	⑤ low quality
⑤ user involvement at the beginning	⑤ —"—
⑥ not iterative	⑥ iterative.

Difference between V-model and RAD model

V-Model	RAD model
① development and testing are not concurrent	① components or functions are developed in parallel.
② User involvement is at beginning	② —11—
③ little flexible	③ more flexible
④ expensive	④ less cost
⑤ testing activities start with the 1st stage.	⑤ testing is done after coding.
⑥ good or high quality product	⑥ low quality product
⑦ not iterative	⑦ iterative

properties of model	waterfall model	incremental model	spiral model	RAD model
① planning in early stage	yes	yes	yes	No
② Returning to an earlier phase	No	yes	yes	yes
③ Handle large project	not appropriate	not appropriate	appropriate	not appropriate
④ Detailed documentation	Necessary	yes but not much.	yes	limited
⑤ cost	low	low	Expensive	low
⑥ Requirement specifications	Beginning	Beginning	Beginning	Time boxed release.
⑦ Flexibility to change	difficult	easy	easy	easy
⑧ User involvement	only at beginning	intermediate	High	only at the beginig
⑨ maintainance	least	Promotes maintainability	typical	Easily maintained
⑩ Duration	Long	very long	long	Short
⑪ Risk involvement	High	low	medium to high risk	low
⑫ framework type	linear	linear + iterative	linear + iterative	linear
⑬ testing	after completion of coding phase	after every iteration	at the end	after completion of coding
⑭ Overlapping phases	No	yes (Parallel)	No	yes
⑮ Re-usability	least possible	To some extent	To some extent	yes
⑯ Time frame	very long	long	long	Short
⑰ working slw availability	at the end of cycle	at the end of every iteration	— " —	at the end of life cycle
⑱ Team size	large team	not large team	large team	small team

Software project management

S/W project management consists of many activities that includes planning of the project, deciding the scope of product, estimation of cost in different terms, scheduling of tasks.

The list of activities are as follows:

1. project planning and tracking
2. project resource management
3. scope management
4. Estimation management
5. Project risk management
6. Scheduling management
7. project communication management
8. ~~Conf~~ configuration management

I] Project planning :-

It is a set of multiple processes or we can say that it a task that performed before the construction of the product starts.

II] Scope management

It describes the scope of the project scope management is important because it clearly defines what would do and what would not. Scope management create the project to contain restricted and quantitative tasks, which may merely be documented and successively avoids price and time overrun.

III] Estimation management :-

This is not only about cost estimation because whenever we start to develop S/W, but

we also figure out their size, efforts, time as well as cost.

if we talk about the size, then line of code depends upon user or SW requirement.

If we talk about effort, we should know about the size of the SW, because based on the size we can quickly estimate how big team required to produce the SW.

If we talk about time, when size and efforts are estimated, the time required to develop the SW can easily determine.

If we talk about cost, it includes all the the elements such as:

① Size of SW
② Quality
③ HW
④ Communication
⑤ Training
⑥ Additional SW and tools
⑦ Skilled manpower

IV] Scheduling management:-

Scheduling management is SW refers to all the activities to complete in the activities to complete in the specified order and within time slotted to each activity. project managers define multiple tasks and arrange them keeping various factors in mind.

For Scheduling, it is compulsory

1. Find out multiple tasks and correlate them.
2. Divide time into units
3. Assign the respective number of work units for every job.

3. Calculate the total time from start to finish.
4. Break down the project into modules.

V] Project Resource Management:-

In SW development, all the elements are referred to as resources for the project. it can be a human resource, productive tools, and libraries.

Resource management includes:-

1. create a project team and assign responsibilities to every team member.
2. Developing a resource plan is derived from the project plan.
3. Adjustment of resources.

VI] project Risk management:-

Risk management consists of all the activities like identification, analyzing and preparing the plan for predictable and unpredictable risk in the project.

Several points show the risks in the project.

1. The experienced team leaves the project, and the new team joins it.
2. changes in requirements.
3. change in technologies and the environment
4. market competition.

VII] Project communication management:-

Communication is an essential factor in the success of the project. It is a bridge between client, organization, team members and as well as other stakeholders of the project such as HW suppliers.

Estimation: —

Estimation of the size of the S/W is an essential parts of S/W project management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used to project size, estimation.

1. lines of code
2. No. of entities in ER diagram
3. Total no. of processes in detailed data flow dia.
4. Function points.

1. Lines of code (LOC): —

As the name suggests, LOC count the total no. of lines of source code in a project.

The units of LOC are: -

KLOC: — Thousand lines of code
NLOC: — Non-comment lines of code
KDSI: — Thousands of delivered source instruction.

The size is estimated by comparing it with the existing systems of the same kind. The experts use it to predict the required size of various components of S/W and then add them to get the total size.

The length of time it takes to solve an issue is measured in LOC. This statistic will differ greatly from one programmer to next. A seasoned programmer can write the same logic in fewer lines than a newbie coder.

Advantage:-
1. University accepted and is used in many models like cocomo.
2. Estimation is closer to the developer's perspective.

Disadvantage:-
1. Different programming languages contain a different number of lines.
2. No proper industry standard exists for this technique in the early stages of the project.

II] No. of entities in ER diagram

ER model provides a static view of the project. it describes the entities and their relationships. The no. of entities in ER model can be used to measure the estimation of the size of the project. The no. of entities depends on the size of the project. This is because more entities needed more classes/ structures then leading to more coding.

Advantage:-
1. Size estimation can be done during the initial stages of planning.
2. The number of entities is independent of the programming technologies used.

Disadvantages.
1. No fixed standards exist. Some entities contribute more project size than others.
2. It is less used in the cost estimation model. Hence, it must be converted to LOC.

III] Total No of processes in detailed data flow diagram.

⇒ Data flow diagram (DFD) represents the Functional View of SW. The model depicts the main processes/ Functions involved in SW and the flow of data between them. Utilization of the no. of functions in DFD to predict SW size. Already existing processes of similar type are studied and used to estimate the size of the process. Sum of the estimated size of each process gives the final estimated size.

Advantages:-

1. It is independent of the programming language.
2. Each major process can be decomposed into it

Disadvantages:-

1. Studying similar kinds of processes to estimate size takes additional time and effort.
2. All SW projects are not required for the construction of DFD.

IV] Function point Analysis:-

In this method, the number of type of functions supported by the SW are utilized to find FPC (Function point count). The steps in function point analysis are:-

1. Count the number of functions of each proposed type.
2. Compute the unadjusted function points (UFP)
3. Find total degree of influence (TDI)
4. Compute Value Adjustment (VAF)
5. Find the Function point count (FPC)

Compute the unadjusted Function points (UFP)
categorise each of the five function types like simple, average, or complex based on their complexity. multiply the count of each function type with its weighting factor and find the weighted sum. The weighting factors for each type based on their complexity are as follows:

Function type	Simple	Average	complex
External i/ps	3	4	6
External o/p	4	5	7
External inquiries	3	4	6
internal logical files	7	10	15
External interface files	5	7	10

[2] Find total degree of influence:-
Use the 14 general characteristics of a system to find the degree of influence of each of them. The sum of all 14 degrees of influence will give the TDI. The range of TDI is 0 to 70. The 14 general characteristics are Data communications, distributed data processing, performance, Heavily used configuration, Transaction rate, on line data entry, end user efficiency, online u/ date, complex processing reuderability, installation ease, operational ease, multiple sites and facilitate change.
Each of the above characteristics is evaluated on a scale of 0-5.

Compute value adjustment Factor (VAF):-
Use the following formula to calculate VAF.

$VAF = (TDI * 0.01) + 0.65$

Find the function point count:-
use the following formula to calculate FPC

$FPC = UFP * VAF.$

Advantages:-

1. It can be easily used in the early stages of project planning.
2. It is independent of the programming language.
3. It can be used to compare different projects even if they use different technologies.

Disadvantage:-

1. It is not good for real time systems and embedded systems.
2. many cost estimation model like COCOMO use LOC and hence FPC must be converted to LOC

COCOMO Model

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981. COCOMO is one of the most generally used SW estimation models in the world. COCOMO predicts the efforts and schedule of a SW product based on the size of the SW.

The necessary steps in this model are:-

1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC)
2. Determine a set of 15 multiplying factors from various attribute of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factor. i.e. multiply the values in step1 and step2.

The inital estimate is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort E, in person months the equation used to of the type is shown below.

$$E_i = a * (KDLOC)^b$$

The value of the constant a and b are depends on the project type.

In COCOMO, projects are categorized into 3 types:-

1. organic
2. Semidetached
3. Embedded.

1. organic :-
A development project can be treated of the organic type, if the project deals with developing a well understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects.
eg. of this type of projects are simple business systems, simple inventory management systems, and data processing systems.

2. Semidetached :-
A development project can be treated with semi detached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed.

3. Embedded :-
A development project is treated to be of an embedded type, if the s/w being developed is strongly coupled to complex H/w. or if the stringent regulations on the operational method exist. eg. ATM, Air Traffic control.

According to Boehm, s/w cost estimation should be done through three stages :.
1. Basic model
2. Intermediate model
3. Detailed model.

1. Basic COCOMO Model :-

The basic cocomo model provide an accurate size of the project parameters. The following expressions give the basic cocomo estimation model.

$$Effort = a_1 * (KLOC)^{a_2} \ PM$$

$$T_{dev} = b_1 * (efforts)^{b_2} \ months$$

where

KLOC is the estimated size of the s/w product indicate in kilo lines of code.

a_1, a_2, b_1, b_2 are constants for each group of s/w products.

Tdev is the estimated time to develop the s/w, expressed in months,

Effort is the total effort required to develop the s/w product, expressed in person months (PM)

Estimation of development effort.

For the 3 classes of s/w products, the formula for estimating the effort based on the code size are shown below:-

organic : Effort = $2.4 (KLOC)^{1.05}$ PM

Semi-detached : Effort = $3.0 (KLOC)^{1.12}$ PM

Embedded : Effort = $3.6 (KLOC)^{1.20}$ PM

Estimation for development time.

For the 3 classes of s/w products, the formulas for estimating the development time based on the effort are given below:-

organic :- $T_{dev} = 2.5 (Effort)^{0.38}$ months

Semi-detached :- $T_{dev} = 2.5 (Effort)^{0.35}$ months

Embedded :- $T_{dev} = 2.5 (Effort)^{0.32}$ months

Example 1:- suppose a project was estimated to be 400 KLOC. calculate the effort and development time for each of the 3 model eg. organic, semi-detached & embedded

→

The basic COCOMO equation takes the form:-

Effort $= a_1 * (KLOC)^{a_2}$ PM

$T_{dev} = b_1 * (\text{effort})^{b_2}$ months

Estimated size of project = 400 KLOC.

i) organic mode:-

$E = 2.4 * (400)^{1.05} = 1295.31$ PM

$D = 2.5 * (1295.31)^{0.38} = 38.07$ PM

ii) semidetached mode

$E = 3.0 * (400)^{1.12} = 2462.79$ PM

$D = 2.5 * (2462.79)^{0.35} = 38.45$ PM

iii) Embedded mode

$E = 3.6 * (400)^{1.20} = 4772.81$ PM

$D = 2.5 * (4772.8)^{0.32} = 38$ PM

200K LOC

Organic $E = 625.05$ PM

$T_{dev} = 28.87$ months

Semi-detached $E = 1133.11$ PM

$T_{dev} = 29.3$ months

embedded $E = 2077.90$ PM

$T_{dev} = 28.81$ months

Example 2:- A project size of 200 KLOC is to be developed. S/w development team has avg. experience on similar type of projects. The project schedule is not very tight. Calculate the Effort, development time, average staff size, and productivity of the project.

→ The Semidetached mode is the most appropriate mode, keeping in view the size, schedule and experience of development time.

Hence

$$E = 3.0(200)^{1.12} = 1133.12 \text{ PM}$$

$$D = 2.5(1133.12)^{0.35} = 29.3 \text{ PM}$$

Average staff size $= \frac{E}{D}$ persons

$$= \frac{1133.12}{29.3}$$

$$= 38.67 \text{ persons}$$

$$\text{Productivity} = \frac{KLOC}{E} = \frac{200}{1133.12}$$

$$= 0.1765 \text{ KLOC/PM}$$

$$P = 176 \text{ LOC/PM}$$

II] Intermediate model:—

The basic COCOMO model considers that the effort is only a function of the no. of lines of code and some constants calculated according to the various S/W systems. The intermediate COCOMO model recognizes these facts and refines the initial estimates obtained through the basic COCOMO model by using a set of 15 cost drivers based on various attributes of S/W Engg.

classification of cost drivers and their attributes:—

(i) Product attributes:—
1. Required S/W reliability extent
2. Size of the application database
3. The complexity of the product.

(ii) H/W attributes:—
1. Run time performance constraints
2. Memory constraints
3. The volatility of the virtual machine environment
4. Required turnabout time.

(iii) Personnel attributes:—
1. Analyst capability
2. S/W Engg. capability
3. Applications experience
4. Virtual machine experience
5. Programming language experience

(iv) Project attributes:—
1. Use of S/W tools
2. Application of S/W Engg. methods
3. Required development schedule

Intermediate COCOMO equation:-

$$E = a_i (KLOC)^{b_i} * EAF$$

$$D = c_i (E)^{d_i}$$

Coefficients for intermediate COCOMO

Project	a_i	b_i	c_i	d_i
organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

III] Detailed COCOMO Model:-

Detailed COCOMO incorporates all qualities of the standard version with an assessment of the cost drivers effort on each method of the SW Engg. process. The detailed model uses various effort multipliers for each cost driver property.

In detailed model uses various efforts multipliers for each cost driver property. In detailed COCOMO, the whole SW is differentiated into multiple modules and then we apply COCOMO in various modules to estimate efforts and then sum the effort.

The 6 phases of detailed COCOMO are:.

1. planning and requirements
2. System Structure
3. complete Structure
4. module code and test
5. integration and test
6. cost constructive model

Basic COCOMO model

$E = a(KLOC)^b$ // Total Effort required for project in man months

$T_{dev} = c(Effort)^d$ // Total time to develop in months

Person required = Effort / Time

E = Effort (M)

T_{dev} = Development time.

Software project	a	b	c	d
organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Given estimated size of project is 300 KLOC

$E = a*(KLOC)^b$

$= 2.4 * (300)^{1.05} = 957.61$ man months.

$T/D = c*(E)^d$

$= 2.5 * (957.61)^{0.38} = 33.95$ months

Avg. Resource Size = E/T

$= \frac{957.61}{33.95} = 28.21$ mans

Productivity of dev = $\frac{KLOC}{E} = \frac{300}{957.61} = 0.3132$ KLOC/mm

$= 313$ LOC/mm

Semi detached

$E = a * (KLOC)^b$

$= 3.0 (300)^{1.12}$

$= 1784.42$ Mm (man months)

$T = c * (E)^d$

$= 2.5 (1784.42)^{0.35}$

$= 34.35$ months (M)

Embedded

$E = a * (KLOC)^b$

$= 3.6 (300)^{1.2}$

$= 3379.46$ MM

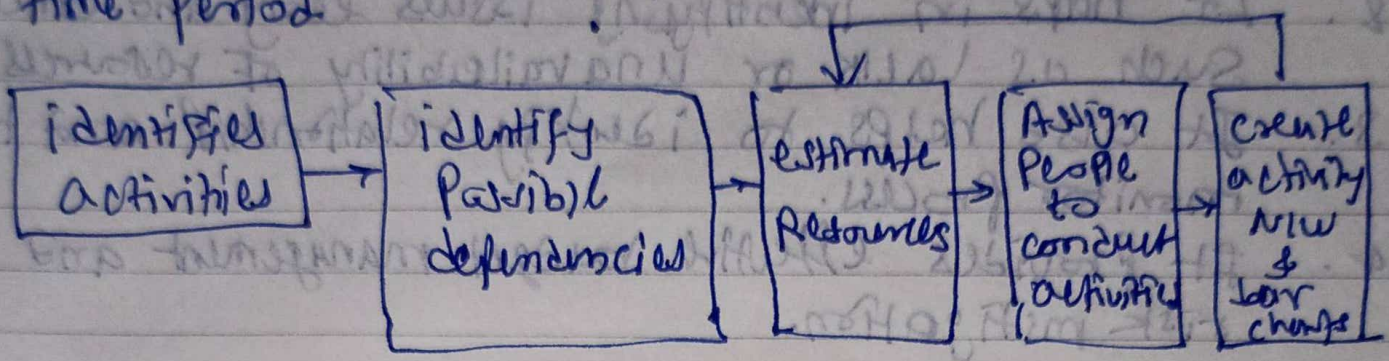
$T = c * (E)^d$

$= 2.5 * (3379.46)^{0.32}$

$= 33.61$ months

Project Scheduling

A schedule in your projects time table actually consists of sequenced activities and milestones that are needed to be delivered under a given time period.



Project Scheduling Process

Project schedule simply means a mechanism that is used to communicate and know about that tasks are needed and has to be done or performed and which organizational resources will be given or allocated to these tasks and in what time duration or time frame work is needed to be performed.

Effective project scheduling leads to success of project, reduce cost, and increased customer satisfaction. Scheduling in project management means to list out activities, deliverables, and milestones within a project that are delivered. It contains more notes than your average weekly planner notes.

The most common and important form of Project schedule is Gantt chart

Advantages of Project Scheduling

1. It simply ensures that everyone remains on same page as far as task gets completed, dependencies and deadlines.
2. It helps in identifying issues early and concerns such as lack or unavailability of resources
3. It also helps to identify relationship and to monitor process.
4. It provides effective budget management and risk mitigation.

Earned Value Management (EVM)

Earned value management (EVM) is possibly one of the most productive technique to measure the performance of a project. A project manager always follows a plan do check act management cycle to ensure all the measures of the project. That's where earned value management helps the project managers to conclude the plan do check - act management cycle.

EVM allows the project manager to find that all the tasks of the project are going on according to the schedule or not, whether the problems occuring are critical or not. EVM not only helps in recoginizing the problems but also provides the approach that is needed to take provides the approach that is needed to ~~take~~ take the project back on ~~track~~ track.

Key practices of EVM includ:-

1. Establish a performance measurement Baseline (PMB):-

The most prominent baseline used for measuring the project in the PMB. The PMB is a measure that the project manager uses to verify against certain benchmarks to get to know where the project currently is.

PMB is based on 3 factors cost, time, & scope. under the baseline, the work is divided into an achievable level and time buget is created for each task to maintain the integrity of PMB throughout the project.

2] Evaluate Performance against the baseline:-
After establishing PMB, it also becomes curtial to evaluate and analyze the project performance against the baseline. To evaluate performance, the project manager tracks the usage of resources, physical work progress, calculates the cost per schedule performance, and findout the concerns in a project to perform the required action during project implementation.

Problems due to the absence of Baseline:-

1. monitoring of progress problem:-
monitoring means that we can compare the projects progress with a plan. In actuality, the baseline is used to compare the actual work with the planned work so that we can evaluate the progress of a project.
for eg.
if a project is planned to be completed in 3 months with a cost of $1000 and after a month the project manager realizes that it is not possible to complete the project with such cost and time; he/she can take the required actions for it.

2. Delays in Schedule:-
If things are not pre-planned and baselined on how much time the project needs then the project would be delayed.
for eg.
if a project should be completed in 3 months and it is being started without any baseline or schedule, no-one knows how long it would takes to complete.

5. Lack of planning of resources:-
IF project didn't baseline on cost, schedule and scope, there will be issues in the planning of sufficient resources that are required for the projects completion.

Earned Value Analysis in project Management:-
Earned Value Analysis (EVA) is one of the key tools and techniques used in project management, to have an understanding of how the project is progressing. EVA implies gauging the progress based on earnings or money. both, schedule and cost are calculated on the basis of EVA.

Features of EVA:-
1. Earned value Analysis is an objective method to measure project management performance in ~~teo~~ terms of scope, time and cost.
2. EVA metrics are used to measure project health and project performance.
3. Earned value Analysis is a quantitative technique for assessing progress as the s/w project team moves through the work tasks, allocated to the project schedule.
4. EVA provides a common value scale for every project task.
5. Total hours to complete the project are estimated and every task is given an earned value, based on its estimated (%) of the total.
6. Earned value is a measure of progress to acess percentage of completeness.

Need for EVA :-

EVA provides different measures of progress for different types of tasks. It is the single way for measuring everything in a project. Provides an early warning single for prompt corrective action. The types of singals can be the following :-

a) Bad news does not age well

Holding on to the bad news does not help. The project manager needs to take an immediate action.

b) Still time to recover

In case, the project is not going as per schedule and get delayed, the situation is needs to be taken care of by finding out the reasons that are causing delay and taking the required corrective action.

c) Timely request for additional funds.

While there is time to recover, the need for additional resources of funds can be escalated with an early warning.

Key Elements of EVA

1. Planned value (PV) The approved cost baseline for the work package. It was ~~early~~ earier known as budgeted cost of work Scheduled (BCWS)
2. Earned value (EV)

The bugeted value of the completed work packages. It used to be known as budgeted cost of work performance at a specified point.

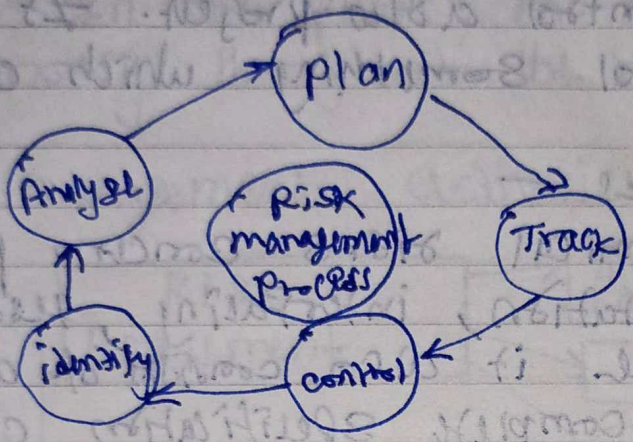
3. Actual cost (AC): The actual cost incurred during the execution of work packages up to a specified point in time. It was previously called actual cost of work performance.

Date: 18
Page:

Risk Management

Risk management is the procell of identifying, assessing and prioritizing the risks to minimize monitor, and control the probability of unfortunate events.

Risk management process:



Risk management practices:—

1. Slw Risk evaluation (SRE)
2. continous Risk management (CRM)
3. Team Risk management (TRM)

A slw project can be concerned with a large variety of risks. In order to be adeft to systematically identify the significant risk which might affect a slw project. it is essential to classify risks into different classes. The project manager can then check which risks from each class are relevant to the project.

There are 3 main classifications of risks which can affect a slw project.

1. Project risks
2. Technical risks
3. Business risks

1. Project risks :-
Project risks concern different forms of budgetary, schedule, personnel, resource and customer related problems. A vital project risk is schedule slippage. Since the s/w is intangible, it is very tough to monitor and control a s/w project. It is very tough to control something which cannot be identified.

2. Technical risks —
Technical risks concern potential method, implementation, interfacing, testing and maintenance issue. it also consists of an ambiguous specification, incomplete specification, changing specification, technical uncertainty.

3. Business risks : —
This type of risk contain risk of building an excellent product that no one need, losing budgetary or personal commitments.

principle of risk management
1. Global perspective :-
In this, we review the bigger system description, design and implementation. we look at the chance and the impact the risk is going to have.

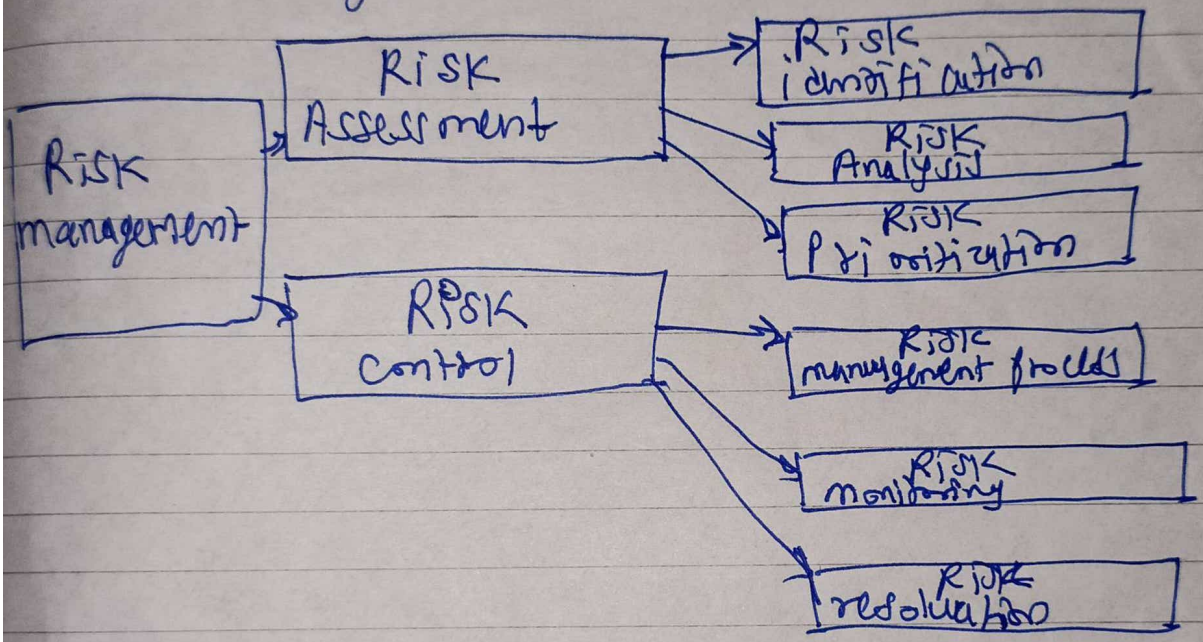
2. Take a forward looking view.
Consider the threat which may appear in the future and create future plans for directing the next events.

3. Open communication :-
This is to allow the free flow of communications between the client and the

Team members so that they have certainty about the risks.

4. Integrated management:—
   In this method risk management is made an integral part of project management.
5. continous process:—
   In this phase, the risks are tracked continously throughout the risk management paradigm.

Risk management activities



Every risk should be split into 2 methods

1. The possibility of a risk coming true denoted by r.
2. The consequence of the issues related to that risk. denoted by S

Based on these two methods, the priority of each risk can be estimated.

$P = r * s$

1. Risk identification
   a. Technology risks
   b. People risks
   c. Organizational risks
   d. Tools risks
   e. Requirement risks
   f. Estimation risks
2. Risk Analysis