Unit - I

Software -

S/w is 1) instructions (computer programs) that when executed provide desired features, functions and performance.

2) data structure that enable the programs to adequately manipulate information and

3) descriptive information in both hard copy and virtual forms that describes the operations and use of programs.

In its most general sense - software is a set of instructions or programs instructing a computer to do specific tasks.

Software Engineering - s/w engineering is

1) "The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software!"

An engineering discipline that is concerned with all aspects of s/w production.

2) The study of approaches as in 1)

s/w characteristics -

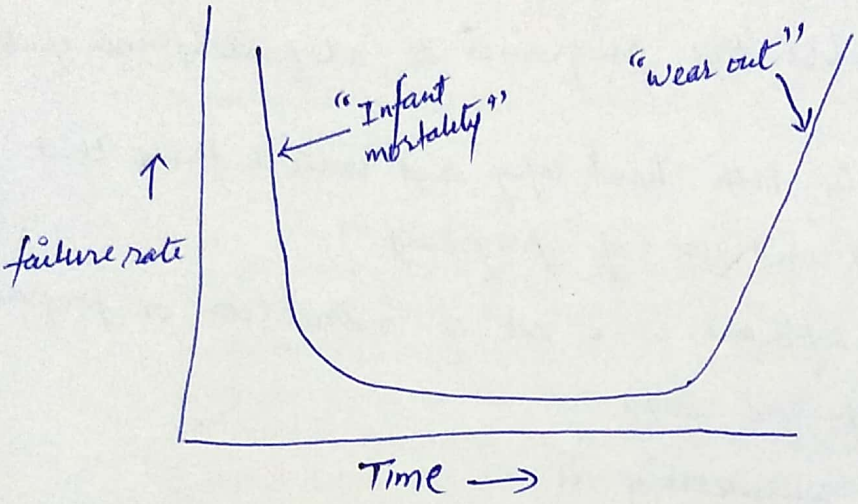
i) "s/w is developed or engineered; it is not manufactured in the classical sense!"

Manufacturing phase for h/w can introduce quality problems that are not nonexistent (or easily corrected) for s/w. Both the activities require the construction of a "product" but the approaches are different.

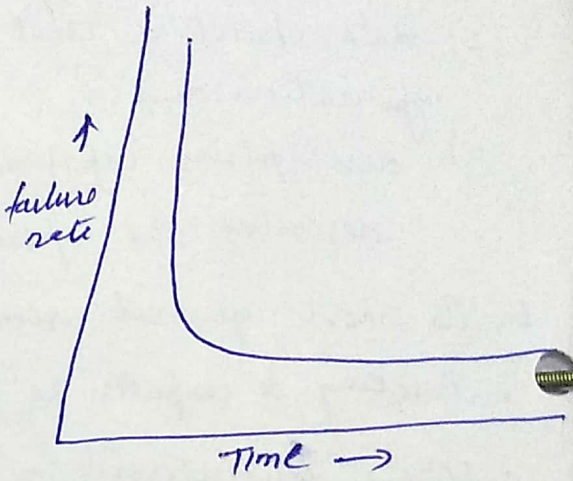
ii) "Software does not wear out"

The h/w exhibits relatively high failure rates early in its life cycle; defects are corrected and the failure rate drops to a steady state level for some period of time. As time passes, however, the failure rate rises again as h/w components suffers from

environmental maladies.
s/w is not susceptible to the environmental maladies that cause h/w to wear out. Undiscovered defects will cause high failure rates early in the life of a program. However these are corrected and the curves flattens.



hardware



software

3) Although the industry is moving toward component based construction most s/w continues to be custom built.
In h/w world, component reuse is a natural part of the engineering process. In the s/w world, it is something that has only begun to be achieved on a broad scale.

[A program is an executable code, which serves some computational purpose. s/w is considered to be collection of executable programming code, associated libraries and documentations.]

[s/w engineering is an engineering branch associated with development of s/w product using well defined scientific principles, methods and procedures.]

Software Application Domains

i) System software
ii) Application s/w
iii) Engineering / Scientific s/w
iv) Embedded s/w
v) Product line s/w
vi) Web applications
vii) Artificial intelligence software

A Layered Technology

s/w development is totally a layered technology. That means to develop a s/w one will have to go from one layer to another. The layers are related and each layer demands the fulfillment of previous layer.

1) A Quality focus – s/w engineering must rest on an organizational commitment to quality. Total quality management and similar philosophies foster a continuous process improvement culture and this culture ultimately leads to the development of increasingly more mature approaches to s/w engineering

2) Process – The foundation for s/w engineering is the process layer. Process defines a framework that must be established for effective delivery of s/w engineering technology. The s/w process forms the basis for management control of s/w projects and establishes the context in which technical methods are applied, work products are produced, milestones are established, quality is ensured and change is properly managed.

3) Methods – s/w engineering methods provides technical how-to's for building s/w. Methods encompass a broad array of tasks that includes communication, requirement analysis, design modeling, program construction, testing and support.

4) Tools – s/w engineering tools provides automated or semi automated support for the process and the methods. Tools are integrated so that information created by one tool can be used by another.

s/w crisis

s/w crisis is a term used in early days of computing science for the difficulty of writing useful and efficient computer program in the required time. The s/w crisis was due to the rapid increases in computer power and the complexity of the problems that could not be tackled. The crisis manifested in several ways.

→ project running over budget
→ project running over time
→ s/w was very inefficient
→ s/w was of very low quality
→ s/w often did not meet requirements
→ project were unmanageable and code difficult to maintain.
→ s/w was never delivered
→ Improper/complex user interface

classes of s/w

Generic s/w – is designed for broad customer market whose requirements are very common, fairly stable and well understood by the s/w engineer.

Customized s/w – is developed for a customer whose domain, environment and requirements are being unique to that customer and can not be satisfied by generic products.

The software Process

→ A process is a collection of activities, actions and tasks that are performed when some work product is to be created.

→ An activity strives to achieve a broad objective and is applied regardless of the application domain, size of the ~~product~~ project, complexity of the effort or degree of rigor with which s/w engineering is to be applied.

→ An action encompasses a set of tasks that produce a major work product.

→ A task focuses on a small, but well defined objective that produces a tangible outcome.

A process framework establishes the foundation for a complete s/w engineering process by identifying a small number of framework activities that are applicable to all s/w projects, regardless of their size or complexity. A generic process framework for s/w engineering encompasses five activities:

Communication - The intent is to understand stakeholders objectives for the project and to gather requirements that helps define s/w features and functions.

Planning - Here a plan to be followed will be created which will describe the technical tasks to be conducted, risks, required resources, work schedule etc.

Modeling - A model will be created to better understand the requirements and design to achieve these requirements.

Construction – This activity combines code generation and the testing that is required to uncover errors in the code.

Deployment – Here the complete s/w or partially completed increment is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

Umbrella Activities

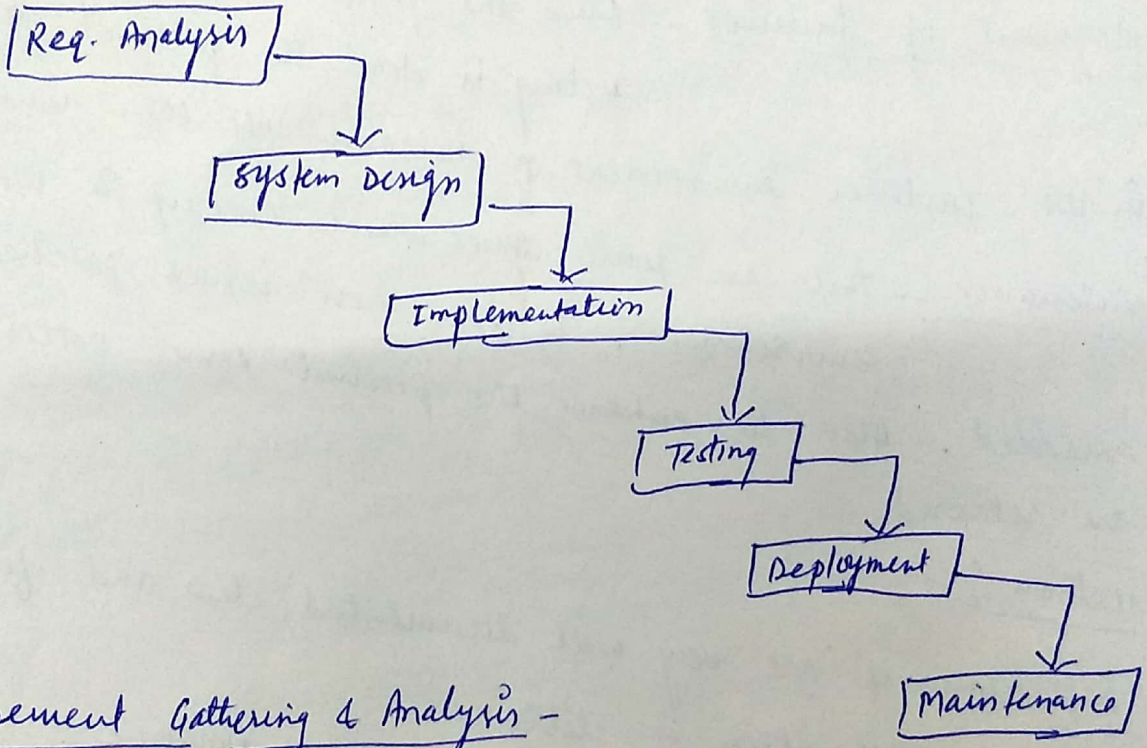
s/w engineering process framework activities are complemented by a number of umbrella activities. Umbrella activities are applied throughout a s/w project and help a s/w team manage and control progress, quality change and risk.

Typical umbrella activities includes:

1) s/w project tracking and control – allows the s/w team to assess progress against the project plan and take any necessary action to maintain the schedule.
2) Risk Management – Assesses the risk that may affect the outcome of the project or the quality of product
3) s/w Quality Assurance – defines and conduct the activities required to ensure s/w quality.
4) Technical Reviews – assesses s/w engineering work products in an effort to uncover and remove errors before they are propagated to the next activity.
5) s/w configuration Management – manages the effects of change throughout the s/w process.
6) Reusability Management – defines criteria for work product reuse and establishes mechanisms to achieve reusable components.

Software Development Life Cycle

Waterfall Model : It is also referred as linear sequential life cycle model. It is very simple to understand and use. In this model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Any phase in the development process begins only if the previous phase is complete.



1) Requirement Gathering & Analysis - All possible requirement of the system to be developed are captured in this phase and documented in a requirement specification document called SRS.

2) System Design — s/w requirement specification (SRS) prepared in previous phase is studied in this phase. A blue print of proposed s/w is prepared in this phase which describes the system in forms of a number of ~~documents~~ diagrams.

3) Implementation – With the inputs from the system design, the system is first developed in small programs called units, which are integrated later on. Each unit is developed and tested for its functionality.

~~4) Testing~~

4) Integration & testing – All the units developed in implementation phase are integrated into a system. Post integration the entire system is tested for any faults and failures.

5) Deployment of failures – Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.

6) Maintenance – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released.

Application :-

a) Requirements are very well documented, clear and fixed.
b) product definition is stable
c) Technology is understood and is not dynamic
d) The project is short

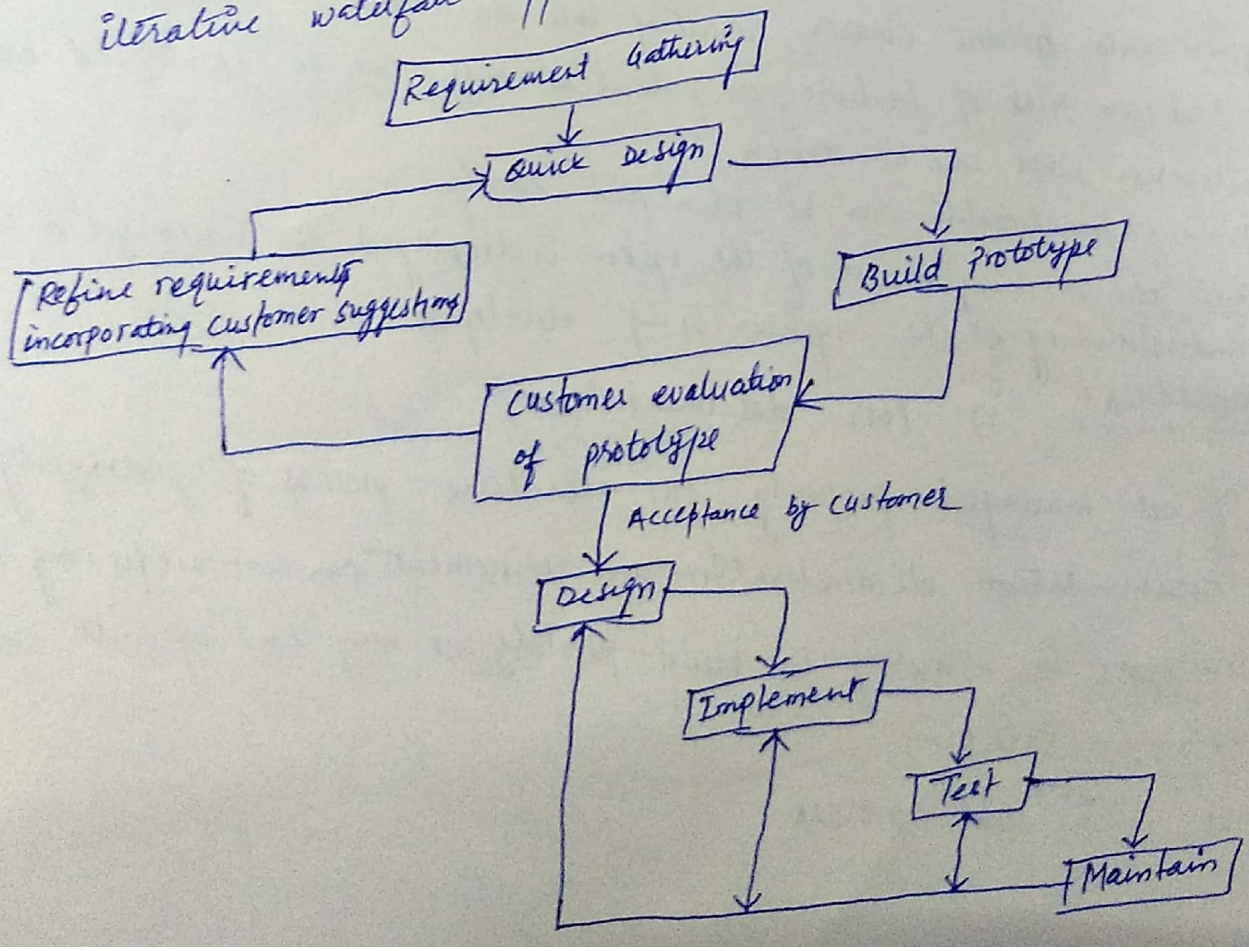
Advantages :-
→ Simple and easy to understand and use.
→ Easy to manage due to high rigidity of the model
→ phases are processed & completed one @ a time
→ well understood milestones
→ Easy to arrange tasks, process & results are well documented.

Disadvantages :-
→ No working s/w is produced until late during the life cycle.
→ High amount of risks and uncertainty
→ poor model for long project
→ Can not accommodate changing requirements
→ Not a good model for complex projects

Prototyping Model

In this model a prototype (an early approximation of final product) is built, tested and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed. A prototype usually exhibits limited functional capability, low reliability and inefficient performance compared to actual s/w.

Product development starts with an initial requirements gathering phase. A quick design is carried out and the prototype is built. The developed prototype is submitted to the customer for his evaluation. Based on customer feedback, the requirements are refined and the prototype is suitably modified. The cycle of obtaining customer feedback and modifying the prototype continues till the customer approves the prototype. The actual system is developed using the iterative waterfall approach.



Prototyping Types

i) Throwaway Prototyping — this type of prototyping uses very little efforts with minimum requirements analysis to build a prototype. Once the actual requirements are understood, the prototype is discarded and the actual system is developed with a much clear understanding of user requirements.

ii) Evolutionary Prototyping - By using this prototyping, the well understood requirements are included in the prototype and the requirements are added as and when they are understood.

iii) Incremental Prototyping - It refers to building multiple functional prototypes of the various sub systems and then integrating all the available prototypes to form a complete system.

Advantages: i) New requirements can be easily added
2) Requirements become clearer resulting into an accurate product.
3) It reduces risk of failure, as potential risks can be identified early and mitigation steps can be taken.
4) Missing functionality can be identified easily.
5) Since the working model of the system is displayed, the users get a better understanding of the system being developed.

Disadvantages: i) Poor documentation
2) If not managed properly, the iterative process of prototyping ~~documentation~~ demonstration and refinement can continue for long duration
3) Developers in a hurry to build prototypes may end up with sub optimal solution.
4) It is a slow process

When to use Iterative Incremental Model

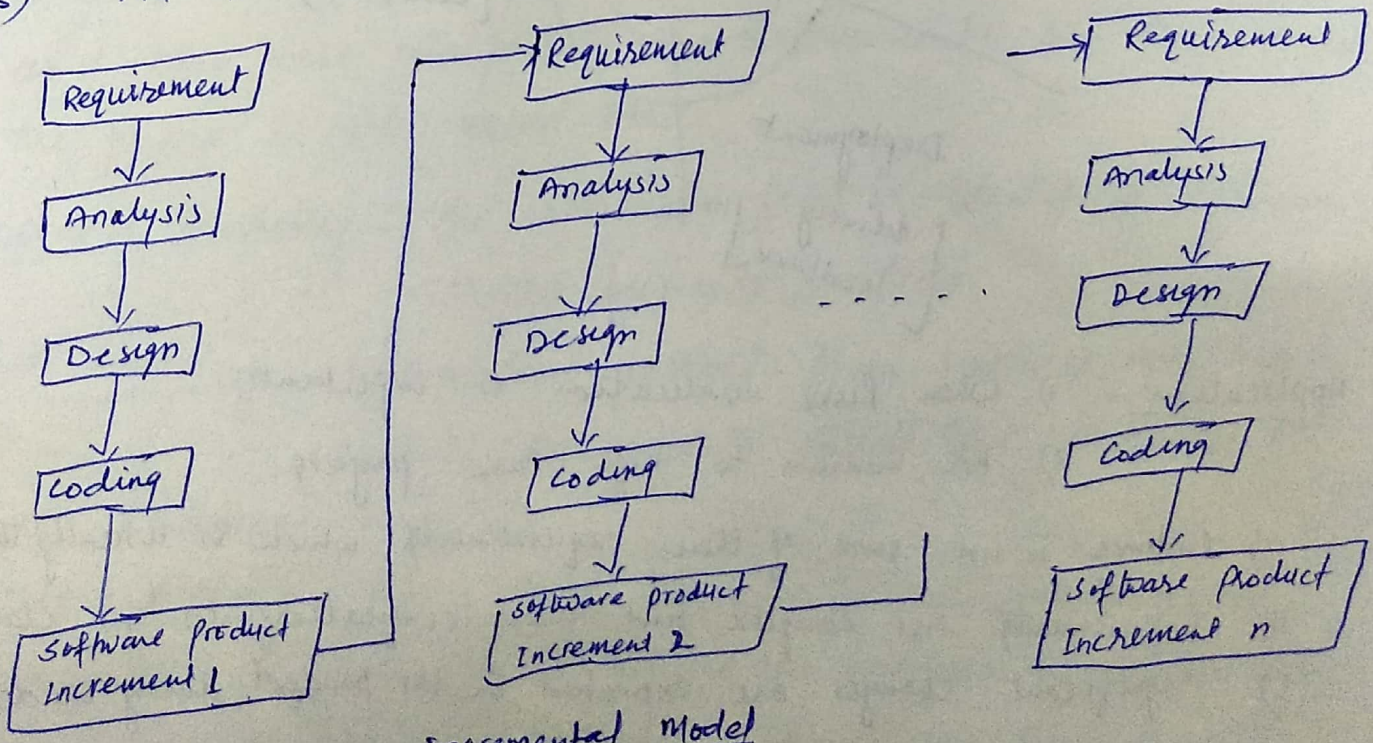
1) The requirements are prioritized.
2) There is a need to get the basic functionality delivered fast.
3) A project has lengthy development schedule
4) The domain is new to the team.
5) Used for developing large applications

Advantages:

1) One can develop prioritized requirements first
2) Initial product delivery is faster
3) Customers gets important functionality early.
4) Requirement change can be easily accommodated.
5) Customer can provide feedback.

Disadvantages:

1) Requires effective planning of iterations
2) Requires efficient design to ensure inclusion of the required functionality and provision for changes later.
3) Total cost of the complete system is higher.
4) Well defined module interfaces are required.
5) Not suitable for small products.

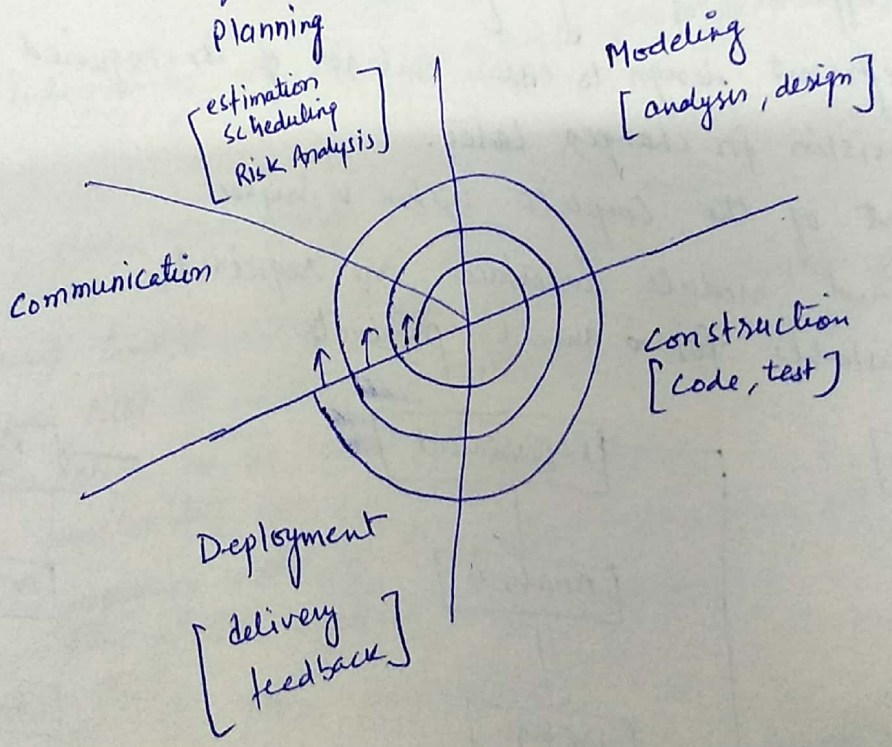


Incremental Model

In Iterative incremental model, initially, a partial implementation of a total system is constructed so that it will be in deliverable state. Increased functionality is added. Defects, if any from the prior delivery are fixed and the working product is delivered. At the end of every iteration, a product increment is delivered.

Spiral Model

spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model along with emphasis on risk analysis. It allows incremental refinement through each iteration around the spiral.



Application - 1) When Risk evaluation is important.
2) For medium to high risk projects.
3) Customer is not sure of their requirements which is usually the case.
4) Requirements are complex and need evaluation to get clarity.
5) Significant changes are expected in the product during the development
6) When project is large

Advantages of Spiral Model

1) Changing requirements can be accommodated
2) User see the system early.
3) Risk can be significantly minimized ie. better risk management
4) Development can be divided into smaller parts.
5) strong approval and documentation control.

Disadvantages of Spiral Model

1) Management is more complex
2) Not suitable for small or low risk projects
3) spiral may go on indefinitely.
4) costly model to use
5) Risk analysis requires highly specific expertise

Rapid Application Development (RAD) Model

In this model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed.
The phase in RAD model are:

Business Modeling - The information flow is identified between various business functions.

Data Modeling - Information gathered from business modeling is used to define data objects that are needed for the business.

Process Modeling - Data objects defined in data modeling are converted to achieve the business information flow and in turn achieve some specific business objective

Application Generation - Automated tools are used to convert process models into code and the actual system.

Testing and Turnover - Test new components and all the interfaces.

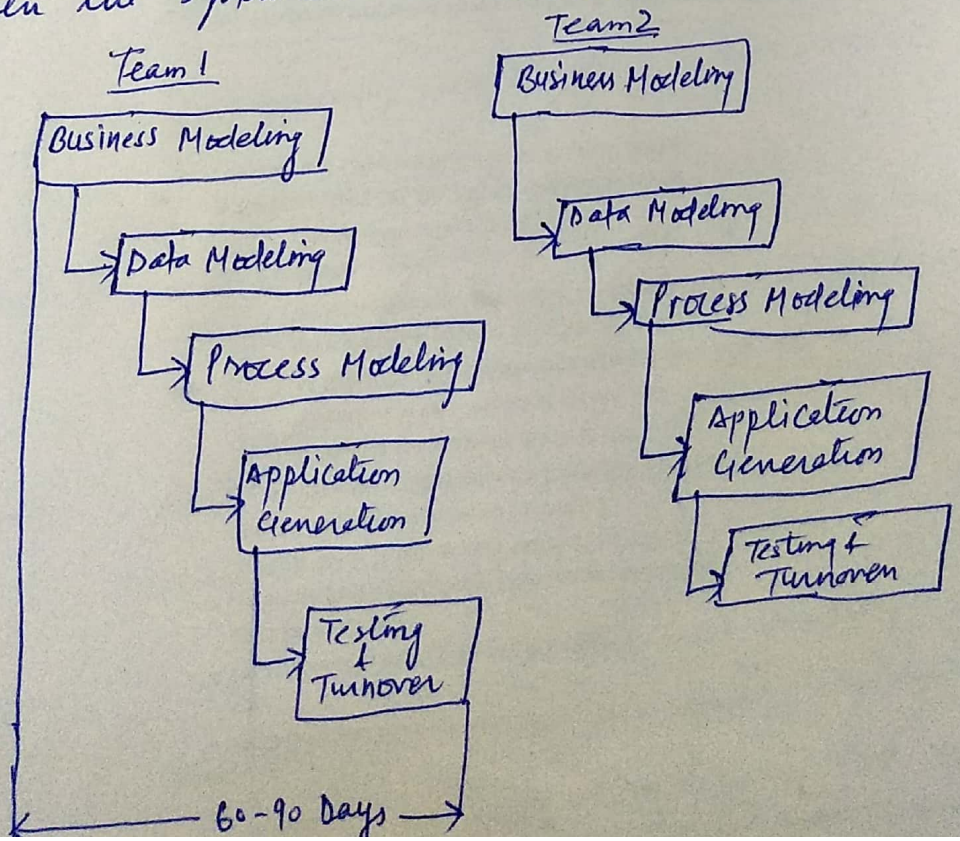
Advantages of RAD Model

1) Reduced development time
2) Increases reusability of components
3) Encourages customer feedback.
4) All functions are modularized so it is easy to work with.

Disadvantages - 1) Requires highly skilled developer/designer.
2) High dependency on modeling skills.
3) Costly model to develop s/w
4) Only system that can be modularized can be built using RAD.

When to Use RAD Model

1) When there is a need to develop system in a short span of time.
2) If there is high availability of skilled designer and budget is high
3) When the system can be modularized.



s/w Quality

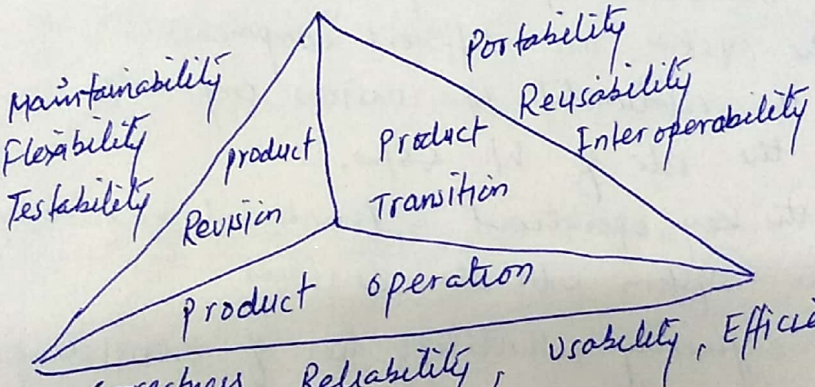
conformance to explicitly stated functional and performance requirements, explicitly documented development standards and implicit characteristics that are expected of all professionally developed s/w.

McCall's Quality factors:

product operation – Its operational characteristics

Product Revision – Its ability to undergo change

Product transition – Its adability to new environments.



Requirement Quality – Ambiguity, Completeness, understandability, volatility, Traceability, Model clarity

Design Quality – Architectural integrity – existence of architectural model
Component completeness – no of component that trace to architectural model
Interface complexity – no of pick to get to a typical function or content.

Code Quality – complexity – cyclomatic complexity, Maintainability, Understandability, Reusability, Documentation.

QC Effectiveness – Resource allocation – staff hour percentage per activity
completion rate – Actual vs budgeted completion time
Testing effectiveness – No of errors found, effort required to correct an error

Why s/w Engineering

To increase s/w productivity and quality
To effectively control s/w schedule and planning
To reduce the cost of s/w development
To meet the customer's needs and requirements
To enhance the conduction of s/w engineering process

Component of s/w Engineering

SE approach has two components, namely system engineering approach and development engineering approach

→ System Engineering approach comprises -
Define the objective of the system.
Define the boundaries of the system.
Factories the system into different components
understand the relationship b/w various components
Understand the role of h/w & s/w.
Identify the key operational & functional requirements.
Discuss the system with the customer.

→ Development engineering methodology has of translating the system requirements as s/w system goals and proceeds to achieve it through a series of steps. Development engineering steps are:

- Requirement definition and specification
- Design solution to deliver the requirement
- Determine the architecture for the delivery of the ~~system~~ solution.
- code development
- s/w testing components
- Integration of system components