Priyadarshini College of Engineering,Nagpur
Department of Computer Technology

Session 2021-2022

Semester: VI-A

Subject: Software Engineering & Project Management BECT308T

Mrs.Rubana A.Khan

- **Syllabus:**
- **UNIT- I**

Introduction: Software Characteristics, Software Engineering- A Layered Technology, Software Process Framework, Software Process Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Process Model, Agile Process Models.

- **UNIT- II**

Software Engineering Principles and Practice: Communication, planning and modeling practices, System engineering and modeling, Business process engineering, Requirements Engineering

- **UNIT- III**

Software Analysis &Design : Modeling Approaches, Data Modeling, Object-Oriented Modelling, Scenario-Based Modeling, Flow-Oriented Modeling, Class-based Modeling, Behavioral Model. Design Engineering Concepts, Design Model, Pattern-Based Software design Design Concepts : Abstraction Architecture, pattern modularity, information hiding, design classes, refactoring.

- **Syllabus:**
- **UNIT -IV**

Software Testing : Testing Fundamentals , Black-Box Testing, White-Box Testing, Unit Testing, Integration Testing, Validation Testing, System Testing, Debugging.

- **UNIT -V**

An overview, Software Quality, A Framework for Product Metrics, Metrics for Analysis & Design Models, Metrics for Source Code, Metrics for Testing & Maintenance. Project management – the management spectrum, Metrics for process & project – Software measurement, Metrics for software quality, Project scheduling.

- **UNIT -VI**

Risk management – Risk strategies, Software risks, Risk identification, Risk refinement, RMMM Quality Management – Quality Concepts, Software Quality Assurance, Software Reviews, Formal Technical Review, Statistical Software Quality Assurance, Software Reliability, Change Management – Software Configuration Management, SCM Repository, SCM Process, Reengineering – Software reengineering, Reverse engineering, Restructuring, Forward Engineering

Course Objectives:

Upon the successful completion of the course, students will be able to:

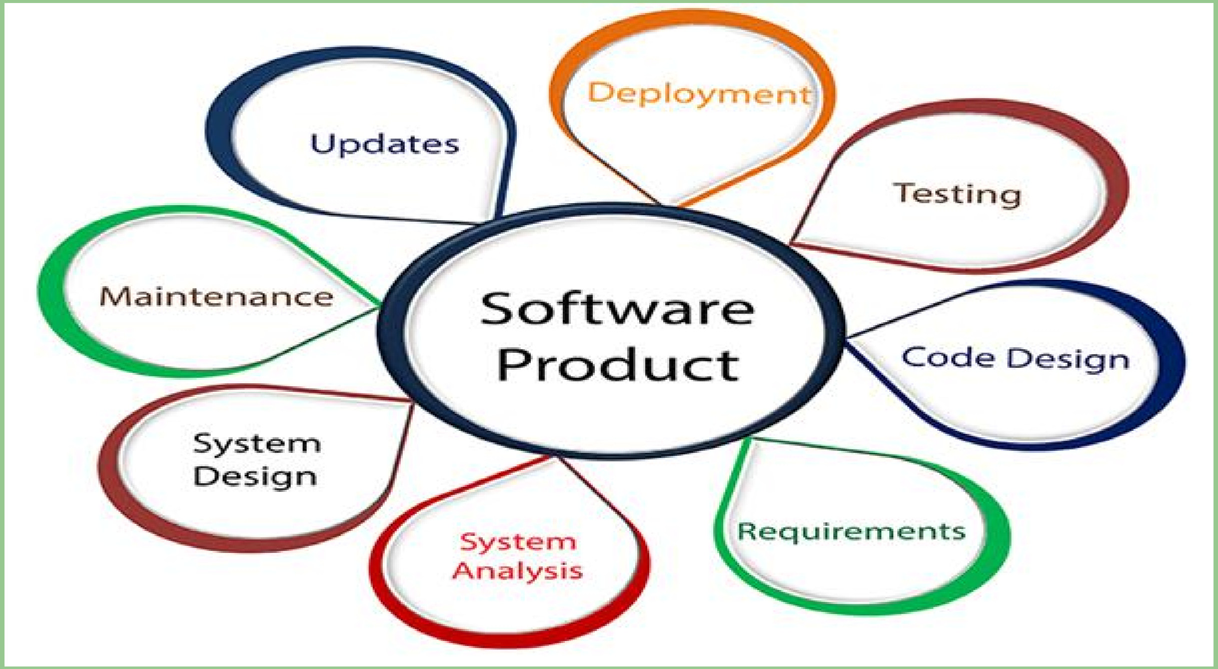
CO Nos.	Course Outcomes	Level (Based on revised Bloom’s Taxonomy)
CO1	To explain evolution and impact of Software Engineering and to demonstrate and compare different software development process models.	I,II
CO2	To list and understand different Software Engineering Principles and practices.	I,II
CO3	To understand , analyze and apply different analysis and design models in software development process.	I,II,III,IV
CO4	To explain and compare different software testing strategies , types and their significance.	I,II
CO5	To understand and apply the concept of Software Quality Assurance and project management.	I,II,III
CO6	To list and analyze different software risk management strategies, software quality management process and to understand software re-engineering concept.	I,II,III,IV

UNIT I:

Introduction: Software Characteristics, Software Engineering- A Layered Technology, Software Process Framework, Software Process Models, Waterfall Model, Incremental Process Models, Evolutionary Process Models, Specialized Process Models, The Unified Process Model, Agile Process Models.

What is Software Engineering?

- The term **software engineering** is the product of two words, **software**, and **engineering**.
- The **software** is a collection of integrated programs.
- Software subsists of carefully-organized instructions and code written by developers on any of various particular computer languages.
- Computer programs and related documentation such as requirements, design models and user manuals.
- **Engineering** is the application of **scientific** and **practical** knowledge to **invent, design, build, maintain**, and **improve frameworks, processes, etc**.



Software Engineering is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures. The result of software engineering is an effective and reliable software product.

Why is Software Engineering required?

Software Engineering is required due to the following reasons:

- To manage Large software
- For more Scalability
- Cost Management
- To manage the dynamic nature of software
- For better quality Management

Need of Software Engineering

The necessity of software engineering appears because of a higher rate of progress in user requirements and the environment on which the program is working.

Huge Programming: It is simpler to manufacture a wall than to a house or building, similarly, as the measure of programming become extensive engineering has to step to give it a scientific process.

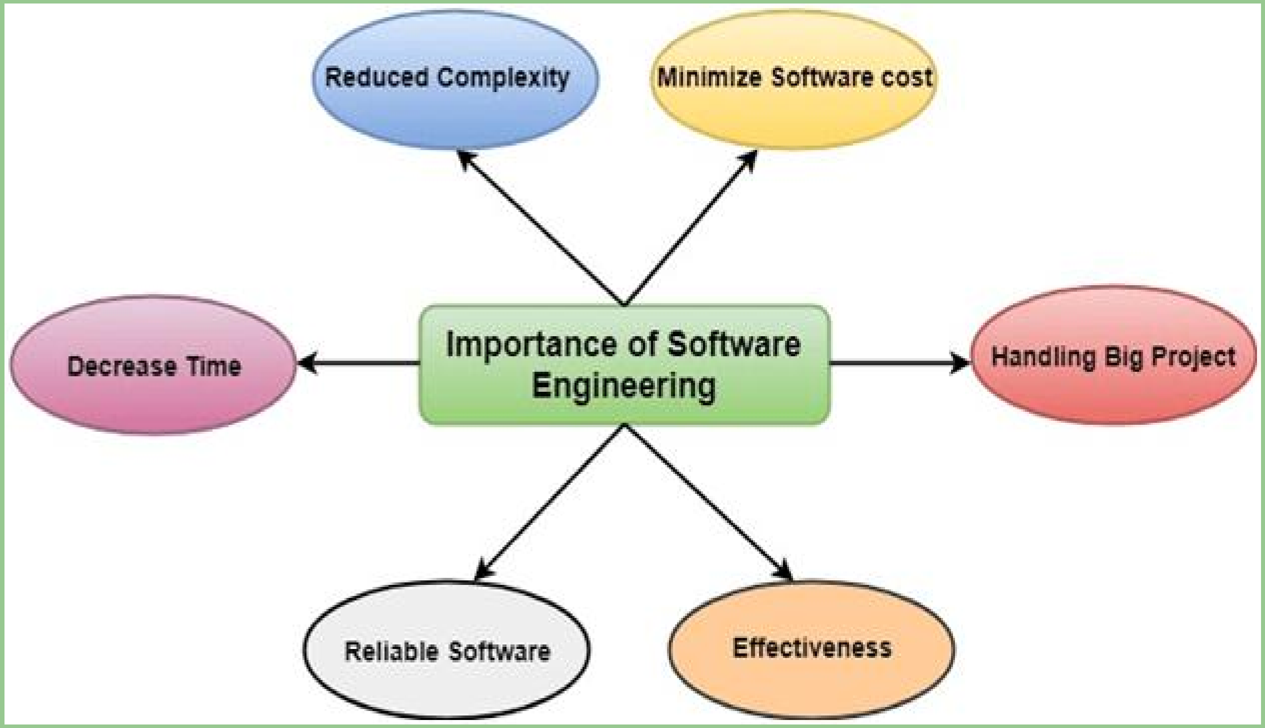
Adaptability: If the software procedure were not based on scientific and engineering ideas, it would be simpler to re-create new software than to scale an existing one.

Cost: As the hardware industry has demonstrated its skills and huge manufacturing has let down the cost of computer and electronic hardware. But the cost of programming remains high if the proper process is not adapted.

Dynamic Nature: The continually growing and adapting nature of programming hugely depends upon the environment in which the client works. If the quality of the software is continually changing, new upgrades need to be done in the existing one.

Quality Management: Better procedure of software development provides a better and quality software product.

Importance of Software Engineering



Importance of Software Engineering

Reduced Complexity

Minimize Software cost

Decrease Time

Handling Big Project

Reliable Software

Effectiveness

The Product

What is it?

– Is the product that software engineers design and build.

– Encompasses programs that executes within a computer of any size and architecture.

Who does it?

– Software engineer and virtually everyone in the industrialized world uses it either directly or indirectly.

Why is it important?

– It affects nearly every aspects of our lives and has become pervasive in our commerce, our culture and our everyday activity.

The Evolving Role of Software

- Plays dual role
- The product
- The vehicle for delivering the product
- Software is an information transformer-
- Produces
- Manages
- Acquires
- Modifies
- Displays

Software acts as the basis for

- The computer (Operating System)
- The communication of information (Networks)
- The creation and control of other programs (Software tools and environments)

The Evolving Role of Software

Software delivers the most important product of our time

- Software transforms personal data
- It manages business information to enhance competitiveness
- Provides a gateway to worldwide information networks
- Provide means of acquiring information in all of its forms

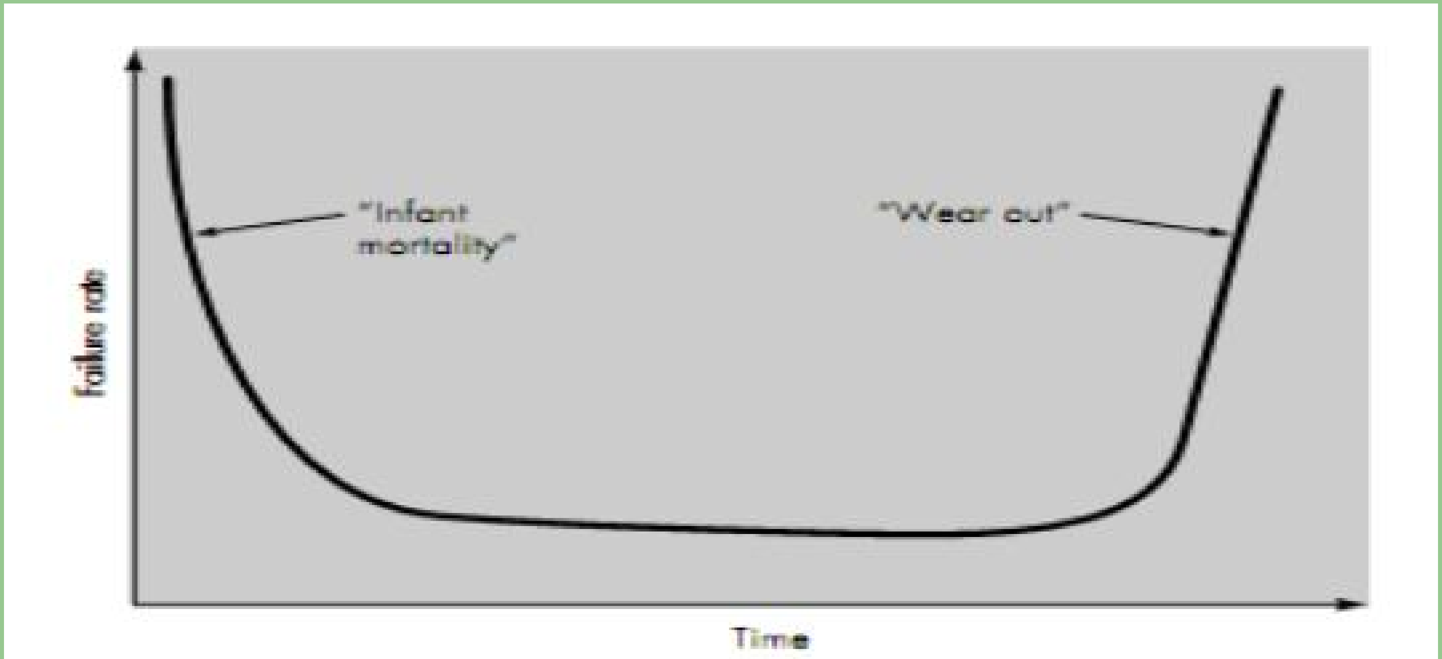
Software

- Software is instruction that when executed provide desired function and performance.
- Is a data structure that enable the programs to adequately manipulate information.
- And documents that describes the operation and use of the program.

Software Characteristics

- Different from hardware.
- Software is developed or engineered, it is not manufactured in the classical sense.
- Software doesn’t **wear out**.
- Although the industry is moving towards component based assembly, most software continues to be custom built.

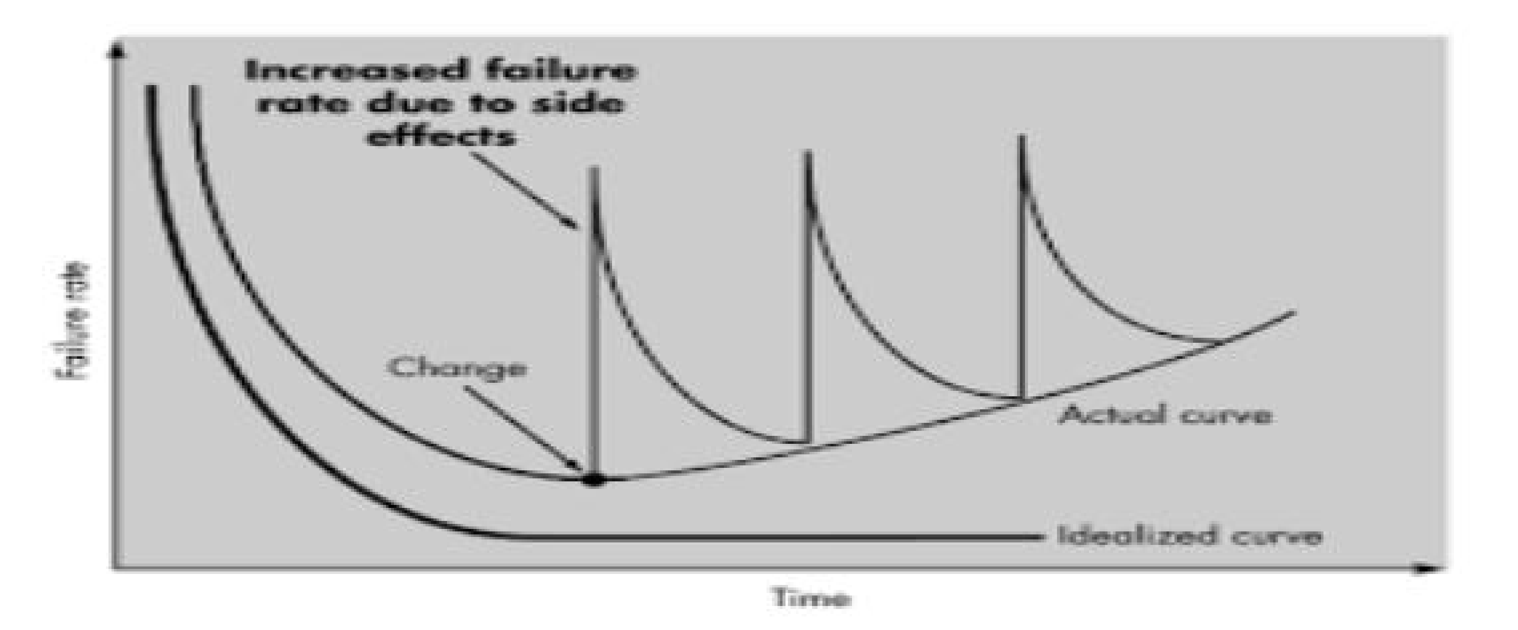
Failure Curve for Hardware



The illustration below depicts failure rate as a function of time for hardware. The relationship, often called the "bathtub curve," indicates the typical failure rate of individual components within a large batch. It shows that in say a batch of 100 products, a relatively large number will fail early on before settling down to a steady rate. Eventually, age and wear and tear get the better of all them and failure rates rise again near the end of the products life. To assist in quality control, many new batches of products are ‘soak’ tested for maybe 24 hours in a hostile environment (temperature/humidity/variation etc.) to pinpoint those that are likely to fail early on in their life, this also highlights any inherent design/production weaknesses

These early failure rates can be attributed to two things • Poor or unrefined initial design. Correcting this, results in much lower failure rates for successive batches of the product. • Manufacturing defects i.e. defects in the product brought about by poor assembly/materials etc. during production. Both types of failure can be corrected (either by refining the design, or by replacing broken components out in the field), which lead to the failure rate dropping to a steady-state level for some period of time. As time passes, however, the failure rates rise again as hardware components suffer from the cumulative effects of dust, vibration, abuse, temperature extremes and many other environmental maladies. Stated simply, "…The hardware begins to wear out."

Software Failure Rates Software is not susceptible to the same environmental problems that cause hardware to wear out. In theory, therefore, the failure rate curve for software should take the form shown below.



Software Applications

System Software

- Written to service other programs
- Real Time Software
- That monitors/ analyzes/ controls real time events
- Business Software
- Business information processing system
- Engineering and Scientific Software
- Characterized by “Number Crunching” algorithms.
- Embedded Software
- Resides in read-only- memory and is used to control products and systems for the consumer

Software Applications

Personal Computer Software

- Word processing, spreadsheets etc..
- Web Based Software
- Web pages retrieved by a browser is a software
- Artificial Intelligence
- Make use of non-numerical algorithms to solve complex problems that are
amenable to computation or straightforward analysis.
- Eg Expert Systems, Pattern Recognition etc….

Categories of Computer Software

Application Domains:

- Consists of standalone programs that solve a specific business need.
- used to control various business applications in real time.
- It helps a computer user to perform specific tasks.
- People use application software according to their needs.
- It is also known as application package.

The importance of Software engineering is as follows:

Reduces complexity: Big software is always complicated and challenging to progress. Software engineering has a great solution to reduce the complication of any project. Software engineering divides big problems into various small issues. And then start solving each small issue one by one. All these small problems are solved independently to each other.

To minimize software cost: Software needs a lot of hardwork and software engineers are highly paid experts. A lot of manpower is required to develop software with a large number of codes. But in software engineering, programmers project everything and decrease all those things that are not needed. In turn, the cost for software productions becomes less as compared to any software that does not use software engineering method.

To decrease time: Anything that is not made according to the project always wastes time. And if you are making great software, then you may need to run many codes to get the definitive running code. This is a very time-consuming procedure, and if it is not well handled, then this can take a lot of time. So if you are making your software according to the software engineering method, then it will decrease a lot of time.

Handling big projects: Big projects are not done in a couple of days, and they need lots of patience, planning, and management. And to invest six and seven months of any company, it requires heaps of planning, direction, testing, and maintenance. No one can say that he has given four months of a company to the task, and the project is still in its first stage. Because the company has provided many resources to the plan and it should be completed. So to handle a big project without any problem, the company has to go for a software engineering method.

Reliable software: Software should be secure, means if you have delivered the software, then it should work for at least its given time or subscription. And if any bugs come in the software, the company is responsible for solving all these bugs. Because in software engineering, testing and maintenance are given, so there is no worry of its reliability.

Effectiveness: Effectiveness comes if anything has made according to the standards. Software standards are the big target of companies to make it more effective. So Software becomes more effective in the act with the help of software engineering.

Software engineering - Layered technology

- Software engineering is a fully layered technology.
- To develop a software, we need to go from one layer to another.
- All these layers are related to each other and each layer demands the fulfillment of the previous layer.

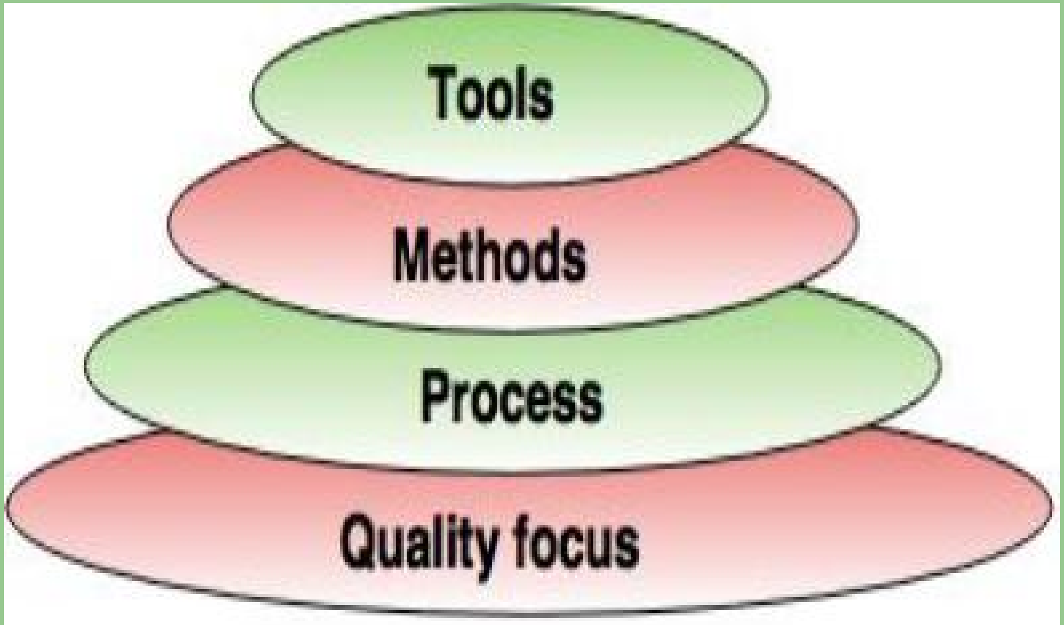


Fig. - Software Engineering Layers

Mrs.Rubana A.Khan

The layered technology consists of:

1. Quality focus

The characteristics of good quality software are: Correctness of the functions required to be performed by the software.

Maintainability of the software

Integrity i.e. providing security so that the unauthorized user cannot access information or data.

Usability i.e. the efforts required to use or operate the software.

2. Process It is the base layer or foundation layer for the software engineering.

The software process is the key to keep all levels together.

It defines a framework that includes different activities and tasks.

In short, it covers all activities, actions and tasks required to be carried out for software development.

3. Methods The method provides the answers of all 'how-to' that are asked during the process.

It provides the technical way to implement the software.

It includes collection of tasks starting from communication, requirement analysis, analysis and design modelling, program construction, testing and support.

4. Tools The software engineering tool is an automated support for the software development.

The tools are integrated i.e the information created by one tool can be used by the other tool.

For example: The Microsoft publisher can be used as a web designing tool.

Software Process Framework:

The process of framework defines a small set of activities that are applicable to all types of projects.

The software process framework is a collection of task sets.

Task sets consist of a collection of small work tasks, project milestones, work productivity and software quality assurance points.

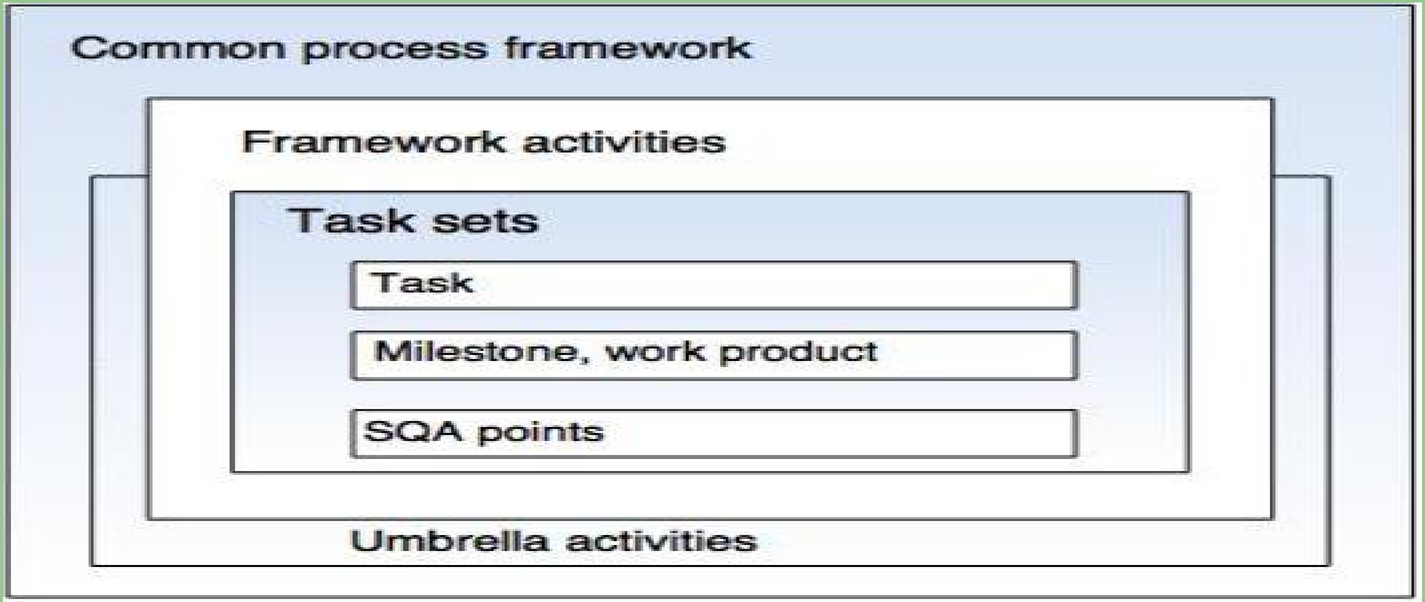


Fig.- A software process framework

Umbrella activities

Typical umbrella activities are:

1. Software project tracking and control In this activity, the developing team accesses project plan and compares it with the predefined schedule.
If these project plans do not match with the predefined schedule, then the required actions are taken to maintain the schedule.
2. Risk management Risk is an event that may or may not occur.
If the event occurs, then it causes some unwanted outcome. Hence, proper risk management is required.
3. Software Quality Assurance (SQA)SQA is the planned and systematic pattern of activities which are required to give a guarantee of software quality.
For example, during the software development meetings are conducted at every stage of development to find out the defects and suggest improvements to produce good quality software.
4. Formal Technical Reviews (FTR)FTR is a meeting conducted by the technical staff.
The motive of the meeting is to detect quality problems and suggest improvements.
The technical person focuses on the quality of the software from the customer point of view.

5. Measurement Measurement consists of the effort required to measure the software.
The software cannot be measured directly. It is measured by direct and indirect measures.
Direct measures like cost, lines of code, size of software etc.
Indirect measures such as quality of software which is measured by some other factor. Hence, it is an indirect measure of software.
6. Software Configuration Management (SCM) It manages the effect of change throughout the software process.
7. Reusability management It defines the criteria for reuse the product.
The quality of software is good when the components of the software are developed for certain application and are useful for developing other applications.
8. Work product preparation and productionI t consists of the activities that are needed to create the documents, forms, lists, logs and user manuals for developing a software.