COURSE CONTENT

SUBECT: SOFTWARE ENGINEERING

UNIT-I

FACULTY: SUBHRAJJIT SAHOO

ASST. PROF. ME DEPT

COURSE CONTENT

INTRODUCTION:

Software Engineering is a framework for building software and is an engineering approach to software development. Software programs can be developed without S/E principles and methodologies but they are indispensable if we want to achieve good quality software in a cost effective manner.
Software is defined as: Instructions + Data Structures + Documents

Engineering is the branch of science and technology concerned with the design, building, and use of engines, machines, and structures. It is the application of science, tools and methods to find cost effective solution to simple and complex problems.

SOFTWARE ENGINEERING is defined as a systematic, disciplined and quantifiable approach for the development, operation and maintenance of software.

The Evolving role of software
The dual role of Software is as follows:

1. A Product- Information transformer producing, managing and displaying information.
2. A Vehicle for delivering a product- Control of computer(operating system),the communication of information(networks) and the creation of other programs.
3. **Software is developed or engineered**, but it is not manufactured in the classical sense.
4. **Software does not wear out,** but it deteriorates due to change.
5. **Software is custom built** rather than assembling existing components.
6. **System software.** System software is a collection of programs written to service other programs
7. **Embedded software**-- resides in read-only memory and is used to control products and systems

THE CHANGING NATURE OFSOFTWARE
The various categories of software are

1. System software
2. Application software
3. Engineering and scientific software
4. Embedded software
5. Product-line software
6. Web-applications
7. Artificial intelligence software

COURSE CONTENT

LEGACY SOFTWARE

Legacy software are older programs that are developed decades ago. The quality of legacy software is poor because it has inextensible design, convoluted code, poor and nonexistent documentation, test cases and results that are not achieved.

As time passes legacy systems evolve due to following reasons:

1. The software must be adapted to meet the needs of new computing environment or technology.
2. The software must be enhanced to implement new business requirements.
3. The software must be extended to make it interoperable with more modern systems or database
4. The software must be rearchitected to make it viable within a network environment.

SOFTWARE MYTHS

Myths are widely held but false beliefs and views which propagate misinformation and confusion. Three types of myth are associated with software:

a) Management myth
b) Customer myth
c) Practitioner's myth

MANAGEMENT MYTHS

a) Myth(1)-The available standards and procedures for software are enough.
b) Myth(2)-Each organization feel that they have state-of-art software development tools since they have latest computer.
c) Myth(3)-Adding more programmers when the work is behind schedule can catch up.
d) Myth(4)-Outsourcing the software project to third party, we can relax and let that party build it.

CUSTOMER MYTHS

a) Myth(1)- General statement of objective is enough to begin writing programs, the details can be filled in later.
b) Myth(2)-Software is easy to change because software is flexible

PRACTITIONER'S MYTH

a) Myth(1)-Once the program is written, the job has been done.
b) Myth(2)-Until the program is running, there is no way of assessing the quality.
c) Myth(3)-The only deliverable work product is the working program
d) Myth(4)-Software Engineering creates voluminous and unnecessary documentation and invariably slows down software development.

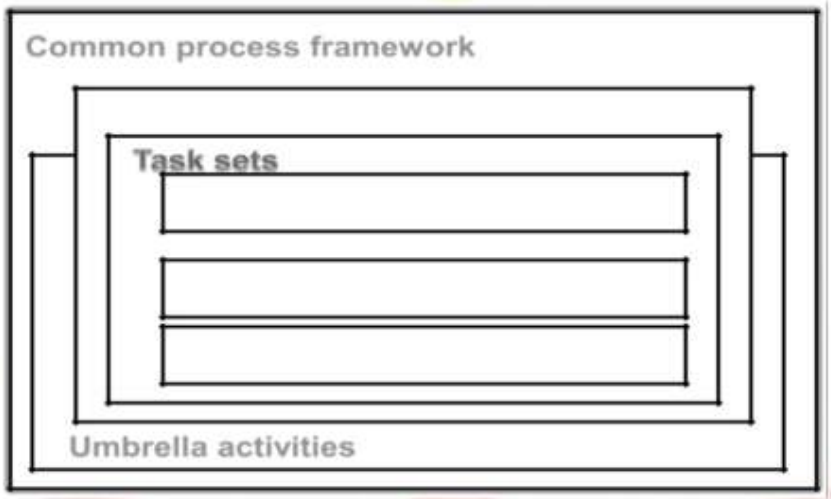
SOFTWARE ENGINEERING-A LAYERED TECHNOLOGY

Fig: Software Engineering-A layered technology

SOFTWARE ENGINEERING - A LAYERED TECHNOLOGY

COURSE CONTENT

A PROCESS FRAMEWORK



A PROCESS FRAMEWORK comprises of :

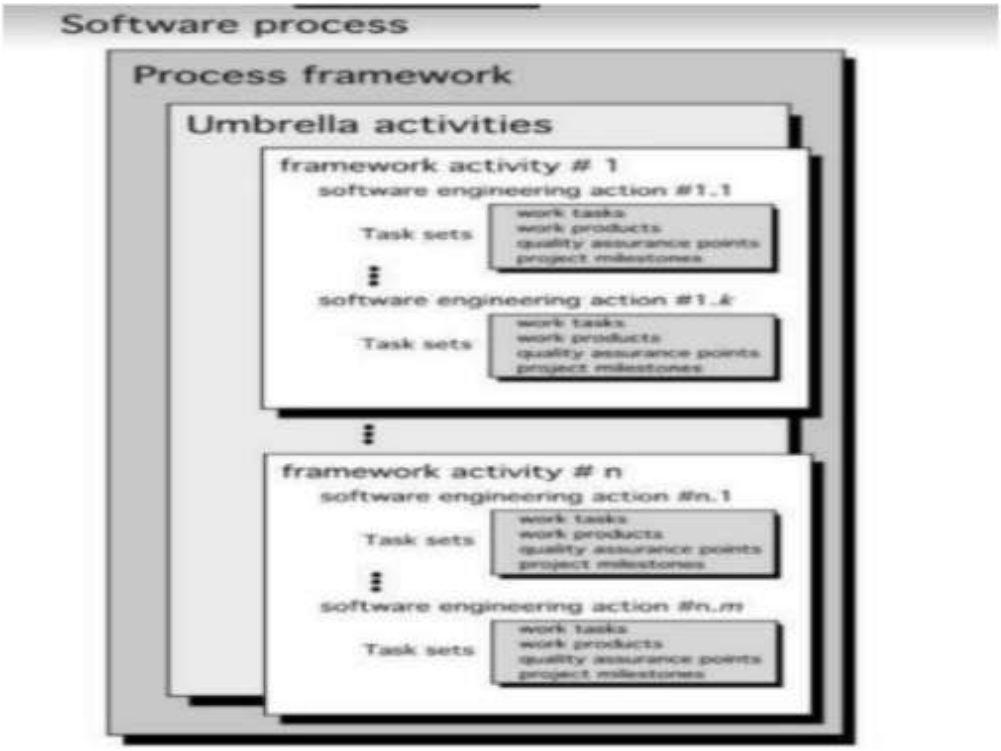
Common process framework Umbrella activities Framework activities Tasks, Milestones, deliverables SQA points

1. Establishes the foundation for a complete software process
2. Identifies a number of framework activities applicable to all software projects
3. Also include a set of umbrella activities that are applicable across the entire software process.

4. Quality focus - Bedrock that supports Software Engineering.
5. Process - Foundation for software Engineering
6. Methods - Provide technical How-to's for building software
7. Tools - Provide semi-automatic and automatic support to methods

8. Communication
9. Planning
10. Modeling
11. Construction
12. Deployment

COURSE CONTENT



A PROCESS FRAMEWORK

Used as a basis for the description of process models Generic process activities

A PROCESS FRAMEWORK

Generic view of engineering complimented by a number of umbrella activities

1. Software project tracking and control
2. Formal technical reviews
3. Software quality assurance
4. Software configuration management
5. Document preparation and production
6. Reusability management
7. Measurement
8. Risk management

CAPABILITY MATURITY MODEL INTEGRATION(CMMI)

1. Developed by SEI(Software Engineering institute)
2. Assess the process model followed by an organization and rate the organization with different levels
3. A set of software engineering capabilities should be present as organizations reach different levels of process capability and maturity.

COURSE CONTENT

CMMI process meta model can be represented in different ways 1.A continuous model
2.A staged model

Continuous model:

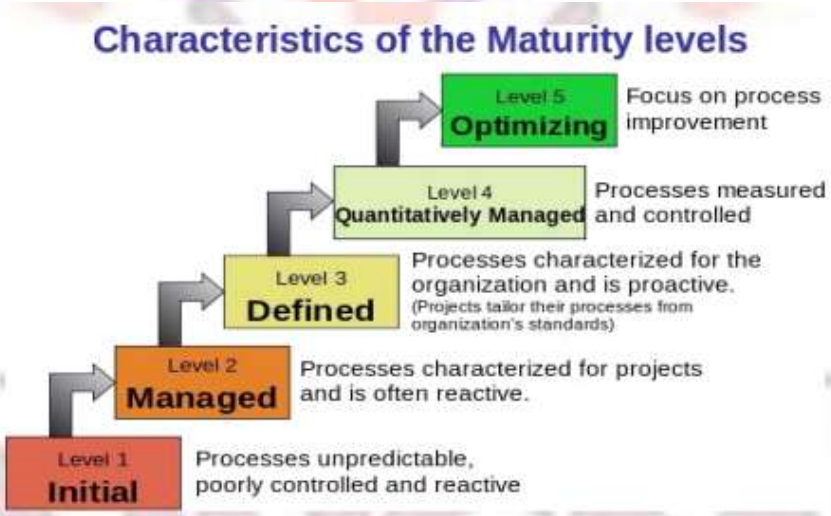
-Lets organization select specific improvement that best meet its business objectives and minimize risk-Levels are called capability levels.

-Describes a process in 2 dimensions

-Each process area is assessed against specific goals and practices and is rated according to the following capability levels.

CMMI

5. Six levels of CMMI
- Level 0:Incomplete
- Level 1:Performed
- Level 2:Managed
- Level 3:Defined
- Level 4:Quantitatively managed
- Level 5:Optimized



CMMI

6. Incomplete -Process is adhoc . Objective and goal of process areas are not known
7. Performed -Goal, objective, work tasks, work products and other activities of software process are carried out
8. Managed -Activities are monitored, reviewed, evaluated and controlled
9. Defined -Activities are standardized, integrated and documented
10. Quantitatively Managed -Metrics and indicators are available to measure the process and quality
11. Optimized - Continuous process improvement based on quantitative feed back from the user

-Use of innovative ideas and techniques, statistical quality control and other methods for process improvement.

COURSE CONTENT

CMMI - Staged model

- This model is used if you have no clue of how to improve the process for quality software.
- It gives a suggestion of what things other organizations have found helpful to work first
- Levels are called maturity levels

PROCESS PATTERNS

Software Process is defined as collection of Patterns.Process pattern provides a template. It comprises of

- Process Template

-Pattern Name -Intent
-Types
-Task pattern -
Stage pattern -
Phase Pattern

- Initial Context
- Problem
- Solution
- Resulting Context
- Related Patterns

PROCESS ASSESSMENT

Does not specify the quality of the software or whether the software will be delivered on time or will it stand up to the user requirements. It attempts to keep a check on the current state of the software process with the intention of improving it.

PROCESS ASSESSMENT

Software Process
Software Process Assessment Software Process improvement Motivates Capability determination

APPROACHES TO SOFTWARE ASSESSMENT

- Standard CMMI assessment (SCAMPI)
- CMM based appraisal for internal process improvement
- SPICE(ISO/IEC 15504)
- ISO 9001:2000 for software

Personal and Team Software Process

Personal software process

- PLANNING
- HIGH LEVEL DESIGN
- HIGH LEVEL DESIGN REVIEW
- DEVELOPMENT
- POSTMORTEM

Personal and Team Software Process

Team software process Goal of TSP

- Build self-directed teams
- Motivate the teams
- Acceptance of CMM level 5 behavior as normal to accelerate software process improvement
- Provide improvement guidance to high maturity organization

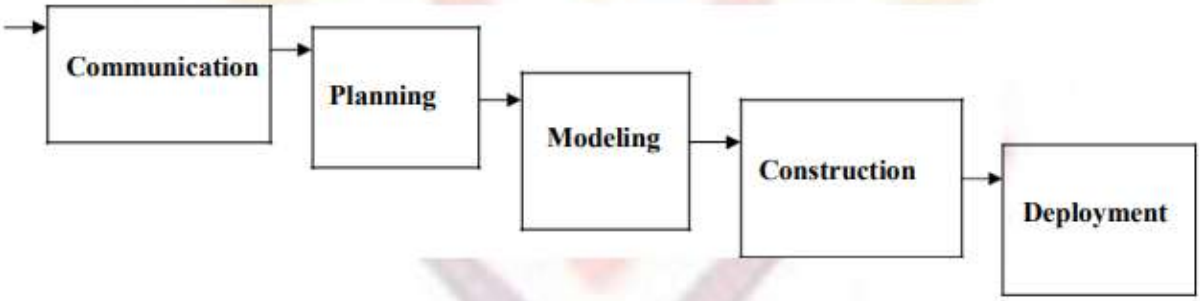
COURSE CONTENT

PROCESS MODELS

- Help in the software development
- Guide the software team through a set of framework activities
- Process Models may be linear, incremental or evolutionary

THE WATERFALL MODEL

- Used when requirements are well understood in the beginning
- Also called classic life cycle
- A systematic, sequential approach to Software development
- Begins with customer specification of Requirements and progresses through planning, modeling, construction and deployment.



This Model suggests a systematic, sequential approach to SW development that begins at the system level and progresses through analysis, design, code and testing

THE INCREMENTAL PROCESS MODEL

- Linear sequential model is not suited for projects which are iterative in nature
- Incremental model suits such projects
- Used when initial requirements are reasonably well-defined and compelling need to provide limited functionality quickly
- Functionality expanded further in later releases
- Software is developed in increments

➢ Communication
➢ Planning
➢ Modeling
➢ Construction
➢ Deployment