SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

Object Oriented Software Engineering

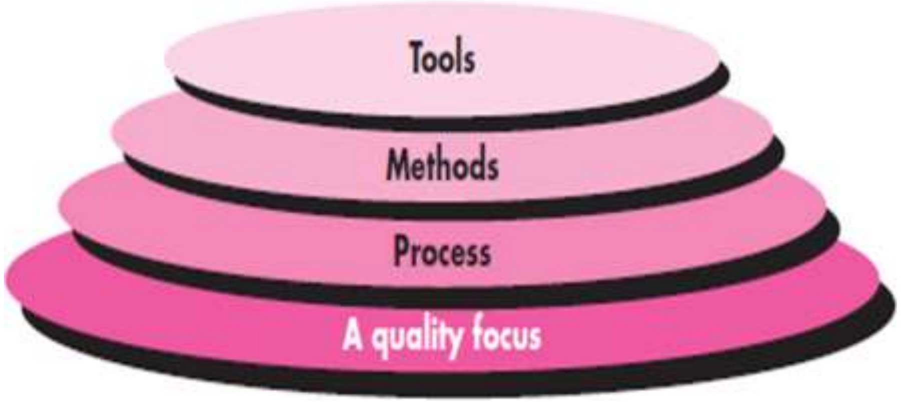
Introduction : Software Engineering & Software Process

Prepared By
R.Vaishnavi.,AP/IT
SNSCE.

Software Engineering

- Software Engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

- **Software engineering is a layered technology.**



Quality focus:

- A disciplined quality management is a backbone of software engineering technology.

Process layer:

- The foundation for software engineering is the *process* layer.
- Process defines a framework that must be established for effective delivery of software engineering technology.

Methods:

- Software engineering *methods* provide the technical how-to’s for building software.
- Methods encompass a broad array of tasks that include communication, requirements analysis, design modeling, program construction, testing, and support.

Tools:

- Software engineering *tools* provide automated or semiautomated support for the process and the methods.
- When tools are integrated so that information created by one tool can be used by another, a system for the support of software development, called *computer-aided software engineering*, is established.

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 (e.g., communication with stakeholders) and is applied regardless of the application domain, size of the project, complexity of the effort, or degree of rigor with which software engineering is to be applied.

➢ An **action** (e.g., architectural design) encompasses a set of tasks that produce a major work product (e.g., an architectural design model).

➢ A **task** focuses on a small, but well-defined objective (e.g., conducting a unit test) that produces a tangible outcome.

A generic process framework for software engineering encompasses five activities:

1) Communication:

- Before any technical work can commence, it is critically important to communicate and collaborate with the customer.
- The intent is to understand stakeholders' objectives for the project and to gather requirements that help define software features and functions.

2) Planning:

- Software project plan—defines the software engineering work by describing the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule.

3) Modeling:

- Software engineers will create models to better understand software requirements and the design that will achieve those requirements.

4) Construction:

- This activity combines code generation and the testing that is required uncovering errors in the code.

5) Deployment:

- The software is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

Umbrella activities:

- ➢ Umbrella activities are applied throughout a software project and help a software team manage and control progress, quality, change, and risk.

Typical umbrella activities include:

1) Software project tracking and control

- allows the software team to assess progress against the project plan and take any necessary action to maintain the schedule.

2) Risk management

- assesses risks that may affect the outcome of the project or the quality of the product.

3) Software quality assurance

- defines and conducts the activities required to ensure software quality.

4) A technical review

- assesses software engineering work products in an effort to uncover and remove errors before they are propagated to the next activity.

5) Measurement

- defines and collects process, project, and product measures that assist the team in delivering software that meets stakeholders' needs.

6) Software configuration management

- manages the effects of change throughout the software process.

7) Reusability management

- defines criteria for work product reuse (including software components) and establishes mechanisms to achieve reusable components.

8) Work product preparation and production

- encompasses the activities required to create work products such as models, documents, logs, forms, and lists.