

UNIT I

SOFTWARE PROCESS AND AGILE DEVELOPMENT

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models –Introduction to Agility-Agile process-Extreme programming-XP Process-Case Study

1. INTRODUCTION TO SOFTWARE ENGINEERING

1.1 SOFTWARE ENGINEERING

The Evolving Role of Software:

Software can be considered in a dual role. It is a product and, a vehicle for delivering a product. As a product, it delivers the computing potential in material form of computer hardware.

Example

A network of computers accessible by local hardware, whether it resides with in a cellular phone or operates in side a main frame computer.

- i) As the vehicle, used to deliver the product. Software delivers the most important product of our time-Information.
- ii) Software transforms personal data, it manages business information to enhance competitiveness, it provides a gateway to worldwide information networks (e.g., Internet)and provides the means for acquiring information in all of its forms.
- iii) Software acts as the basis for operating systems, networks, software tools and environments.

Software Characteristics

Software is a logical rather than a physical system element. Therefore, software hascharacteristicsthatareconsiderablydifferentthanthoseofhardware:

1. Software is developed or engineered; it is not manufactured in the classical sense.

- Although some similarities exist between the envelopment and hardware manufacture, the two activities are fundamentally different.

□ In both activities, high quality is achieved through good design, but the manufacturing phase for hard ware can introduce quality problems that are nonexistent (or easily corrected) for software. Software doesn't "wear out."

1.1.2. Software Application Domains

The following categories of computer software present continuing challenges for software engineers.

a) System software:

- System software is a collection of programs written to service of their programs.
- Example: compilers, editors, and file management utilities, operating system components, drivers, telecommunications processors, process largely indeterminate data.

b) Real-time software:

Elements of real-time software includes

- a data gathering component that collects and formats information from an external environment
- an analysis component that transforms information as required by the application
- a control/output component that responds to the external environment
- a monitoring component that coordinates all other components so that real-time response (typically ranging from 1 millisecond to 1 second) can be maintained.

c) Business software:

- Business information processing is the largest single software application area.
-

Example: payroll, accounts receivable/payable, inventory.

Applications in this area restructure existing data in a way that facilitates business operations or management decision making. In addition to conventional data processing application, business software applications also encompass interactive computing

- Example: point-of-sale transaction processing.

d) Engineering and scientific software:

- This is the software using "number crunching"
- algorithms.

Example: System simulation, computer-aided design.

e) Embedded software:

- Embedded software resides in read-only memory and is used to control products and systems for the consumer and industrial markets.

SOFTWARE ENGINEERING

In order to build software that is ready to meet the challenges and it must recognize a few simple realities:

- It follows that a concerted effort should be made to understand the problem before a software solution is developed.
- It follows that design becomes a pivotal activity.
- It follows that software should exhibit high quality. It follows that software should be maintainable.

These simple realities lead to one conclusion: software in all of its forms and across all application domains should be engineered.

- Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.
- Software engineering encompasses a process, methods for managing and engineering software, and tools.

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 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., 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 process framework establishes the foundation for a complete software engineering process by identifying a small number of framework activities that are applicable to all software projects, regardless of their size or complexity.
- ✓ In addition, the process framework encompasses a set of **umbrella activities** that are applicable across the entire software process. A generic process framework for software engineering encompasses five activities:

The five generic process frame work activities:

a)Communication:

- The intent is to understand stakeholders' objectives for the project and to gather requirements that help define software features and functions.

b) 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.

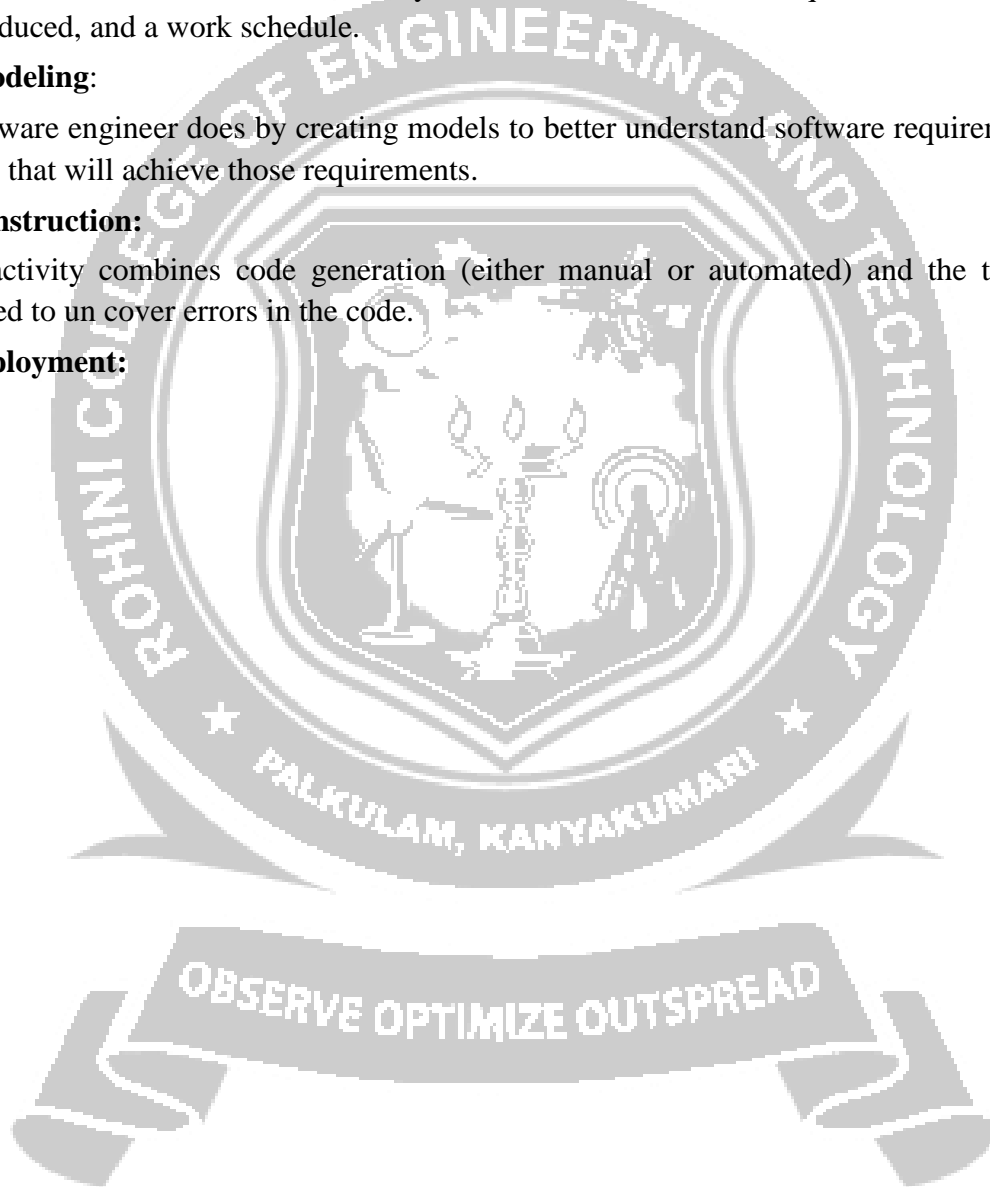
c) Modeling:

- A software engineer does by creating models to better understand software requirements and the design that will achieve those requirements.

d) Construction:

- This activity combines code generation (either manual or automated) and the testing that is required to un cover errors in the code.

e) Deployment:



- Software engineering process framework activities are complemented by a number of umbrella activity.
- In general, 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:
 - i) **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.
 - ii) **Risk management**—assesses risks that may affect the out come of the project or the quality of the product.
 - iii) **Software quality assurance**—defines and conducts the activities required to ensure software quality.
 - iv) **Technical reviews**—access software engineering work products in an effort to uncover and remove errors before they are propagated to the next activity.
 - v) **Measurement**—defines and collects process, project, and product measures that assist the teaming delivering software that meets stakeholders’ needs; can be used in conjunction with all other frame work and umbrella activities.
 - vi) **Software configuration management**—manages the effects of change throughout the software process.
 - vii) **Reusability management**—defines criteria for work product reuse (including software components) and establishes mechanisms to achieve reusable components.
 - viii) **Work product preparation and production**—encompasses the activities required to create work products such as models, documents, logs, forms, and lists.



OBSERVE OPTIMIZE OUTSPREAD