Software Engineering

Lecture 3

Mr. Noor Ul Arfeen

Generic Process Framework

Communication

- Involves communication among the customer and other stake holders; encompasses requirements gathering

Planning

- Establishes a plan for software engineering work; addresses technical tasks, resources, work products, and work schedule

Modelling (Analyse, Design)

- Encompasses the creation of models to better under the requirements and the design

Generic Process Framework

Construction (Code, Test)

- Combines code generation and testing to uncover errors

Deployment

- Involves delivery of software to the customer for evaluation and feedback

Umbrella Framework Activities

Umbrella activities are applied throughout a software project and help a software team to manage and control progress, quality, change, and risk. Typical umbrella activities include:

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

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

Software quality assurance - defines and conducts the activities required to ensure software quality

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

Umbrella Framework Activities

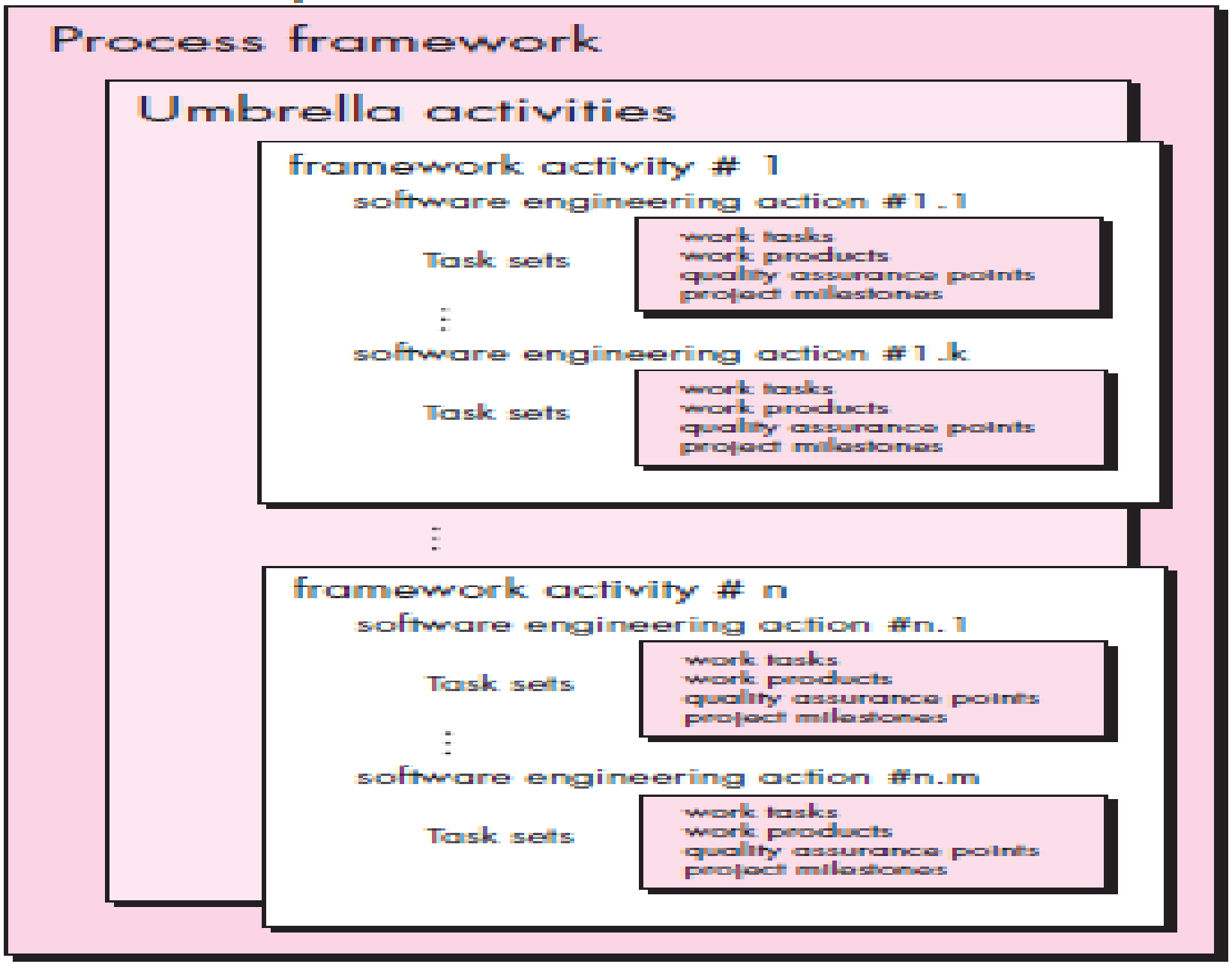
Measurement - defines and collects process, project, and product measures that assist the team in delivering software that meets stakeholders' needs; can be used in conjunction with all other framework and umbrella activities

Software configuration management - manages the effects of change throughout the software process

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

Work product preparation and production - encompasses the activities required to create work products such as models, documents, logs, forms, and lists

Software process



Process Flow

Process Flow describes how the framework activities and the actions and tasks that occur within each framework activity are organized with respect to sequence and time

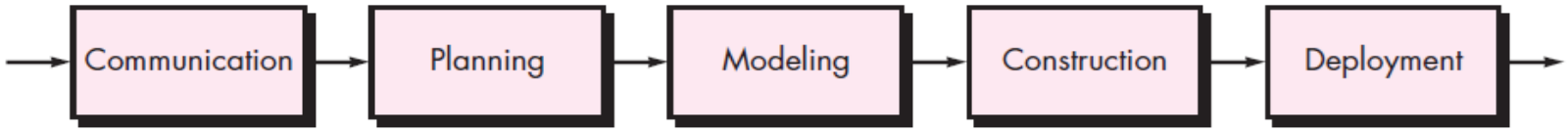
A **linear process flow** executes each of the five framework activities in sequence, beginning with communication and culminating with deployment

An **iterative process flow** repeats one or more of the activities before proceeding to next

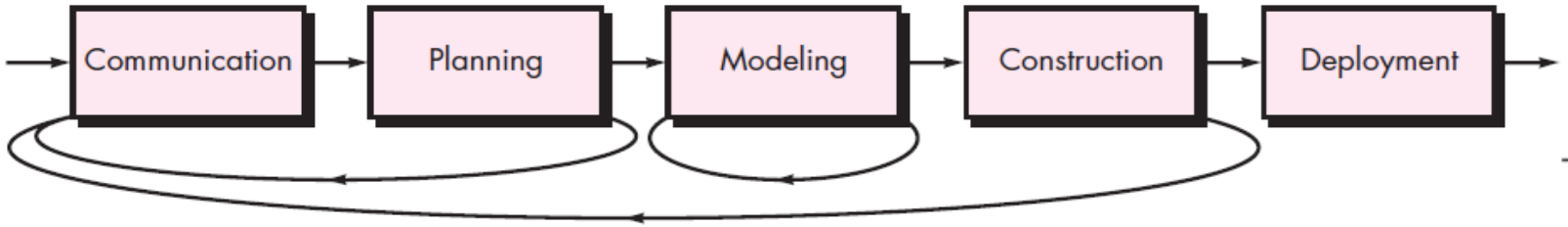
An **evolutionary process flow** executes the activities in a “circular” manner. Each circuit through the five activities leads to a more complete version of the software

A **parallel process flow** executes one or more activities in parallel with other activities (e.g., modeling for one aspect of the software might be executed in parallel with construction of another aspect of the software)

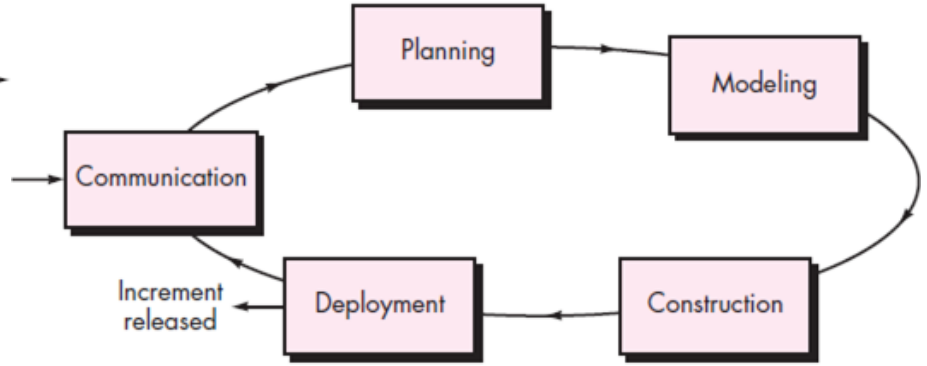
Process Flow



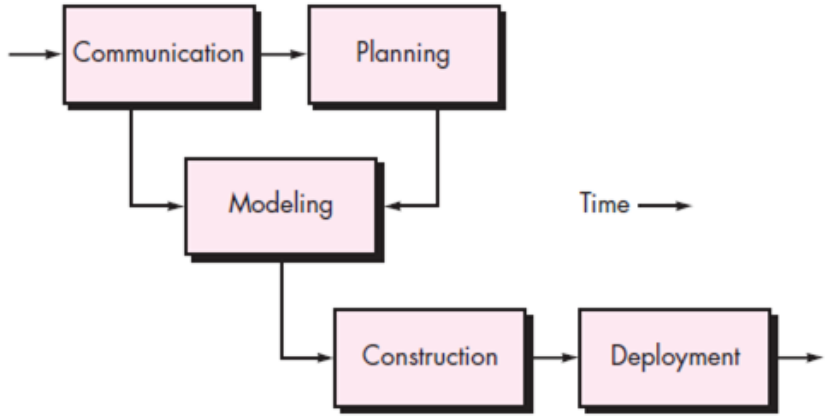
(a) Linear process flow



(b) Iterative process flow



(c) Evolutionary process flow



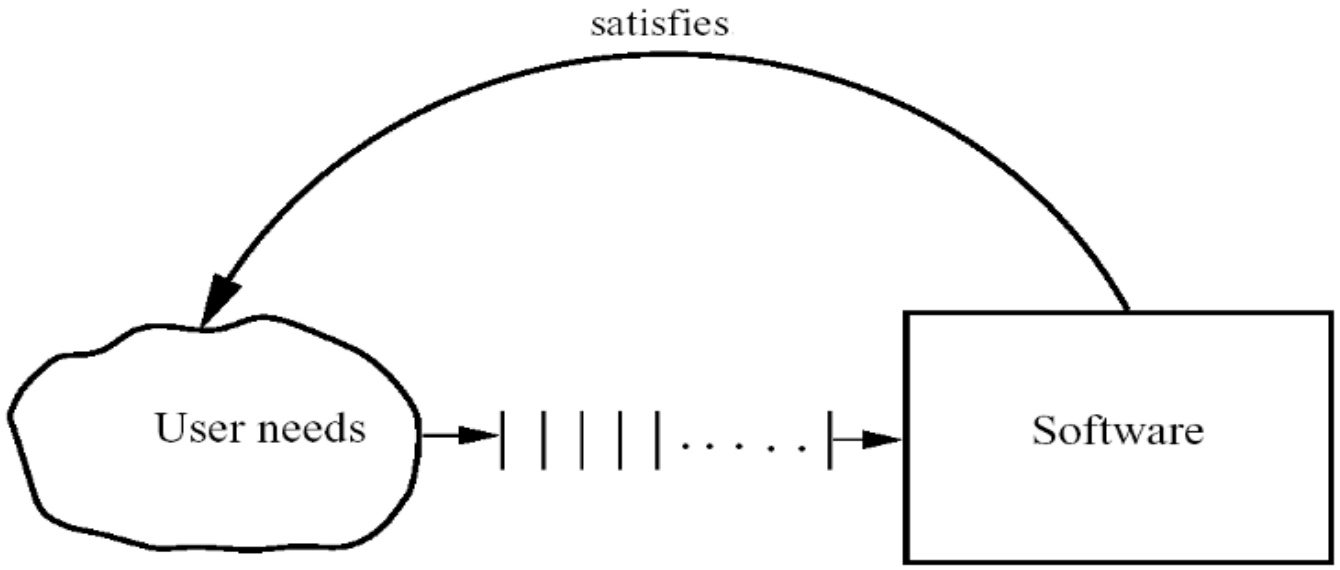
(d) Parallel process flow

Software Process Models

A software process model represents all the activities required to make a software product

Without using the process models, the development of a software product would not be in a systematic and disciplined manner

There must be clear understanding among the team members about when and what to do, Otherwise it would lead to project failure



Waterfall Model

The earliest software development model (Royce, 1970)

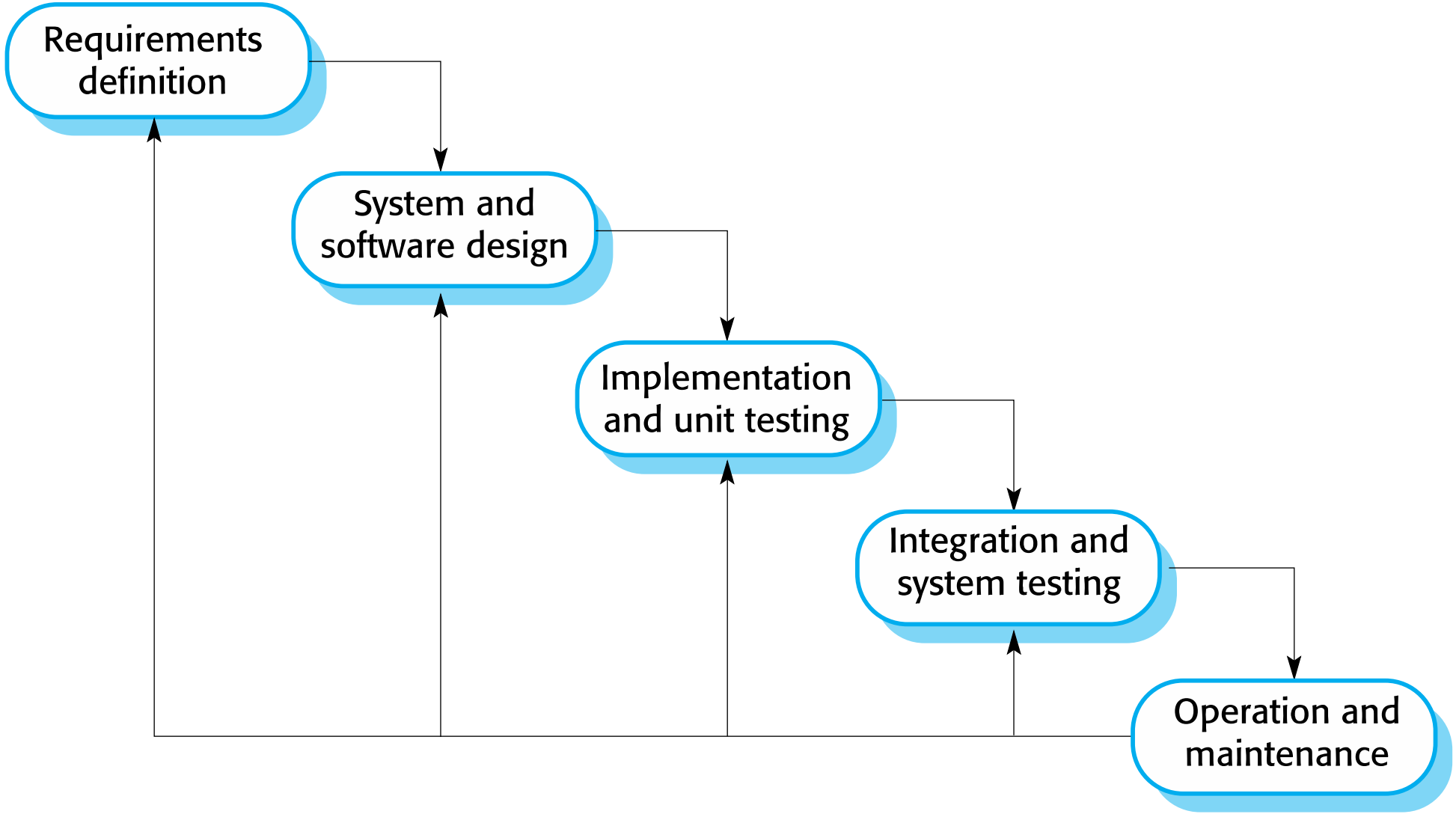
In waterfall model, whole process of software development is divided into separate phases

In a waterfall model, each phase must be completed before the next phase can begin

Waterfall model illustrates the software development process in a linear sequential flow

In this model, the phases do not overlap

Waterfall Model



Waterfall Model

Requirements analysis and definition: The system's services, constraints, and goals are established by consultation with system users. They are then defined in detail and serve as a system specification

System and software design: The systems design process allocates the requirements to either hardware or software systems. It establishes an overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships

Implementation and unit testing: During this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification

Waterfall Model

Integration and system testing: The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer

Operation and maintenance: Normally, this is the longest life-cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors that were not discovered in earlier stages of the life cycle, improving the implementation of system units, and enhancing the system's services as new requirements are discovered

Waterfall Model

Advantages

- Simple and easy to understand and use
- Clearly defined stages
- Phases are processed and completed one at a time
- Works well for smaller projects

Disadvantages

- Cannot accommodate changing requirements
- No backtracking
- Poor model for long projects

Website - Key Teams & Their Roles

Product Manager (Gather requirements from customer)

Design Team (Web Designer and UX Designer)

Development Team (Front-end Team, Back-end Team and Full Stack Developers)

Quality Assurance (QA) Team (test website and identify any bugs)

DevOps and IT Operations Team (manage server infrastructure, deployment processes)

Content Team (Content Writers and Content Editors)

SEO and Digital Marketing Team (optimize website for search engine)

Customer Support Team (assistance to users, answer their questions and address issues)

Legal and Compliance Team (legal requirements such as privacy policies and terms of use)