



# Software Processes

Ashima Tyagi  
Assistant Professor  
School of Computer Science & Engineering

# Outline

- Software Processes
- Framework Activities
- Umbrella Activities
- SDLC
- Process Flow

# Software Process

A software 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. Ex: Gathering and documenting what the software should do.
- **Actions** are **smaller, specific steps** performed within an activity. They focus on achieving a particular goal or outcome within that phase. Ex: Preparing a Software Requirements Specification document.
- **Tasks** are the **smallest units of work**, often performed by individual team members. They are specific, measurable, and actionable. Ex: Write a use case for "User Login."

## Process Framework

- 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.

## Software process

### Process framework

#### Umbrella activities

##### framework activity # 1

software engineering action #1.1

Task sets

⋮

software engineering action #1.k

Task sets

work tasks  
work products  
quality assurance points  
project milestones

work tasks  
work products  
quality assurance points  
project milestones

⋮

##### framework activity # n

software engineering action #n.1

Task sets

⋮

software engineering action #n.m

Task sets

work tasks  
work products  
quality assurance points  
project milestones

work tasks  
work products  
quality assurance points  
project milestones

# Framework Activities

Framework activities are the **core, essential activities** in the software process.

A generic process framework for software engineering encompasses five activities:

- Communication
- Planning
- Modeling
- Construction
- Deployment

These five generic framework activities can be used during the development of small, simple programs, the creation of large Web applications, and for the engineering of large, complex computer-based systems. The details of the software process will be quite different in each case, but the framework activities remain the same.

- **Communication.** Before any technical work can commence, it is critically important to **communicate and collaborate with the customer** (and other stakeholders) The intent is to understand stakeholders' objectives for the project and to gather requirements that help define software features and functions.
- **Planning.** Any complicated journey can be simplified if a map exists. A software project is a complicated journey, and the planning activity creates a **“map” that helps guide the team** as it makes the journey. The map—called **a 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.



- **Modeling.** Whether you're a landscaper, a bridge builder, an aeronautical engineer, a carpenter, or an architect, you work with models every day. You create a "sketch" of the thing so that you'll understand the big picture—what it will look like architecturally, how the constituent parts fit together, and many other characteristics. If required, you refine the sketch into greater and greater detail in an effort to better understand the problem and how you're going to solve it. A software engineer does the same thing by creating models to better understand software requirements and the design that will achieve those requirements.
- **Construction.** This activity combines code generation (either manual or automated) and the testing that is required to uncover errors in the code.
- **Deployment.** The software (as a complete entity or as a partially completed increment) is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

# Umbrella Activities

Umbrella activities are **supportive activities** that span across the entire software process to **enhance the quality, manageability, and maintainability** of the project. Typical umbrella activities include:

- Software project tracking and control
- Risk management
- Software quality assurance
- Technical reviews
- Measurement
- Software configuration management (effects of changes)
- Reusability management
- Work product preparation and production

# SDLC

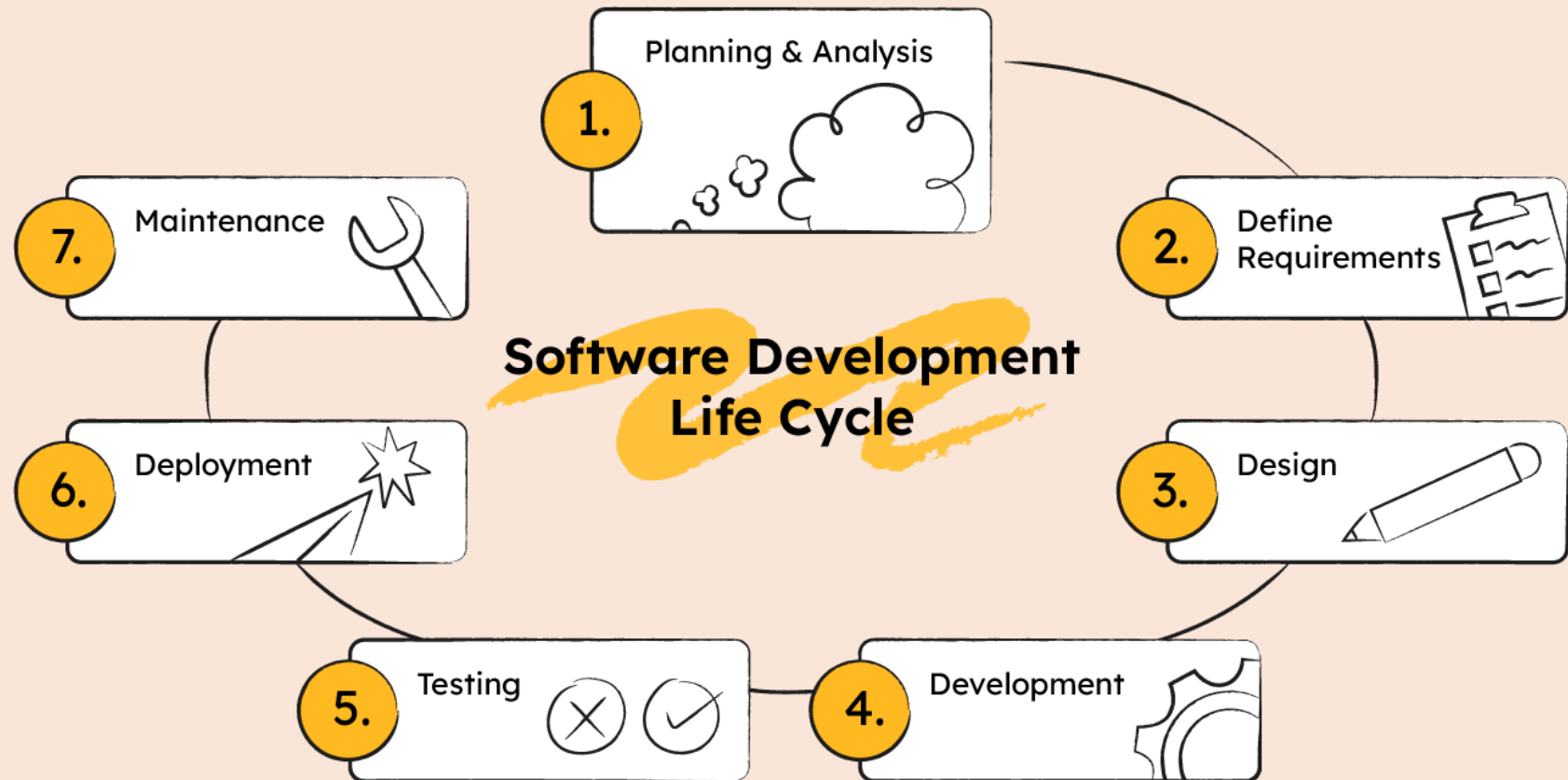
Software development life cycle (SDLC) is a structured process that is used to **design, develop, and test good-quality software**.

SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step.

## Goal of SDLC:

- The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user's requirements.
- SDLC in software engineering models outlines the plan for each stage so that each stage of the software development model can perform its task efficiently to deliver the software at a low cost within a given time frame that meets users' requirements.

## 7 Phases of SDLC



theproductmanager.com



## 1. Planning & Analysis

The first phase of the SDLC is the project planning stage where you are gathering business requirements from your client or stakeholders. This phase is when you evaluate the feasibility of creating the product, revenue potential, the cost of production, the needs of the end-users, etc.

To properly decide what to make, what not to make, and what to make first, you can use a feature prioritization framework that takes into account the value of the software/update, the cost, the time it takes to build, and other factors.

Once it is decided that the software project is in line with business and stakeholder goals, feasible to create, and addresses user needs, then you can move on to the next phase.

## 2. Define Requirements

This phase is critical for converting the information gathered during the planning and analysis phase into clear requirements for the development team.

This process guides the development of several important documents: a software requirement specification (SRS) or product specification, a Use Case document, and a Requirement Traceability Matrix document.

### 3. Design

The design phase is where you put pen to paper—so to speak. The original plan and vision are elaborated into a software design document (SDD) that includes the system design, programming language, templates, platform to use, and application security measures. This is also where you can flowchart how the software responds to user actions.

In most cases, the design phase will include the development of a prototype model. Creating a pre-production version of the product can give the team the opportunity to visualize what the product will look like and make changes without having to go through the hassle of rewriting code.



## 4. Development

The actual development phase is where the development team members divide the project into software modules and turn the software requirement into code that makes the product.

This SDLC phase can take quite a lot of time and specialized development tools. It's important to have a set timeline and milestones so the software developers understand the expectations and you can keep track of the progress in this stage.

In some cases, the development stage can also merge with the testing stage where certain tests are run to ensure there are no critical bugs.

Keep in mind, different types of product development software will have different specialties so you'll want to pick the one that suits you best.

## 5. Testing

Before getting the software product out the door to the production environment, it's important to have your quality assurance team perform validation testing to make sure it is functioning properly and does what it's meant to do. The testing process can also help hash out any major user experience issues and security issues.

In some cases, software testing can be done in a simulated environment. Other simpler tests can also be automated.

## 6. Deployment

During the deployment phase, your final product is delivered to your intended user. You can automate this process and schedule your deployment depending on the type.

For example, if you are only deploying a feature update, you can do so with a small number of users (canary release). If you are creating brand-new software, you can learn more about the different stages of the software release life cycle (SRLC).

## 7. Maintenance

The maintenance phase is the final stage of the SDLC if you're following the waterfall structure of the software development process. However, the industry is moving towards a more agile software development approach where maintenance is only a stage for further improvement.

In the maintenance stage, users may find bugs and errors that were missed in the earlier testing phase. These bugs need to be fixed for better user experience and retention. In some cases, these can lead to going back to the first step of the software development life cycle.

The SDLC phases can also restart for any new features you may want to add in your next release/update.

# Process Flow

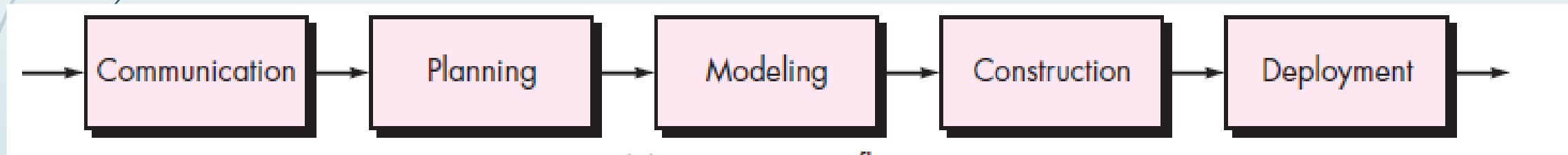
- Process flow—describes how the framework activities and the actions and tasks that occur within each framework activity are organized with respect to sequence and time.

There are different process flows:

1. Linear process flow
2. Iterative process flow
3. Evolutionary process flow
4. Parallel process flow

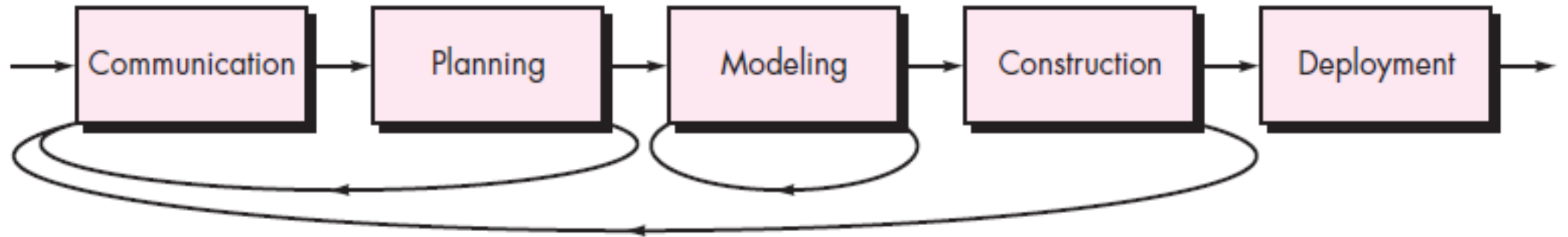
## 1. Linear Process Flow:

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



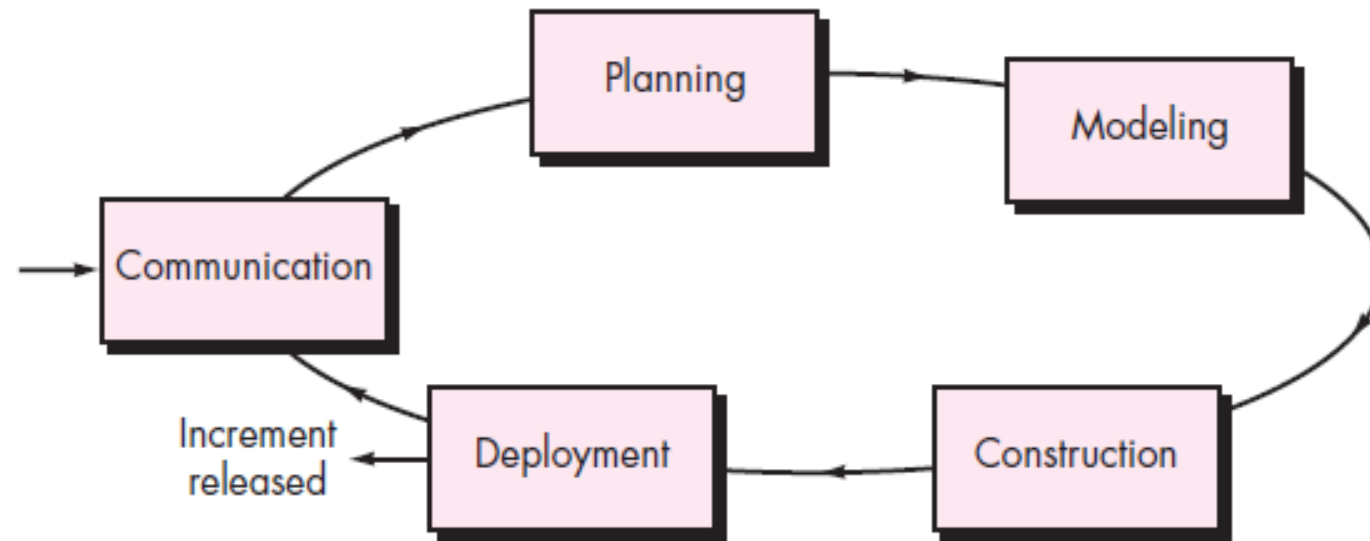
## 2. Iterative Process Flow:

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



### 3. Evolutionary Process Flow:

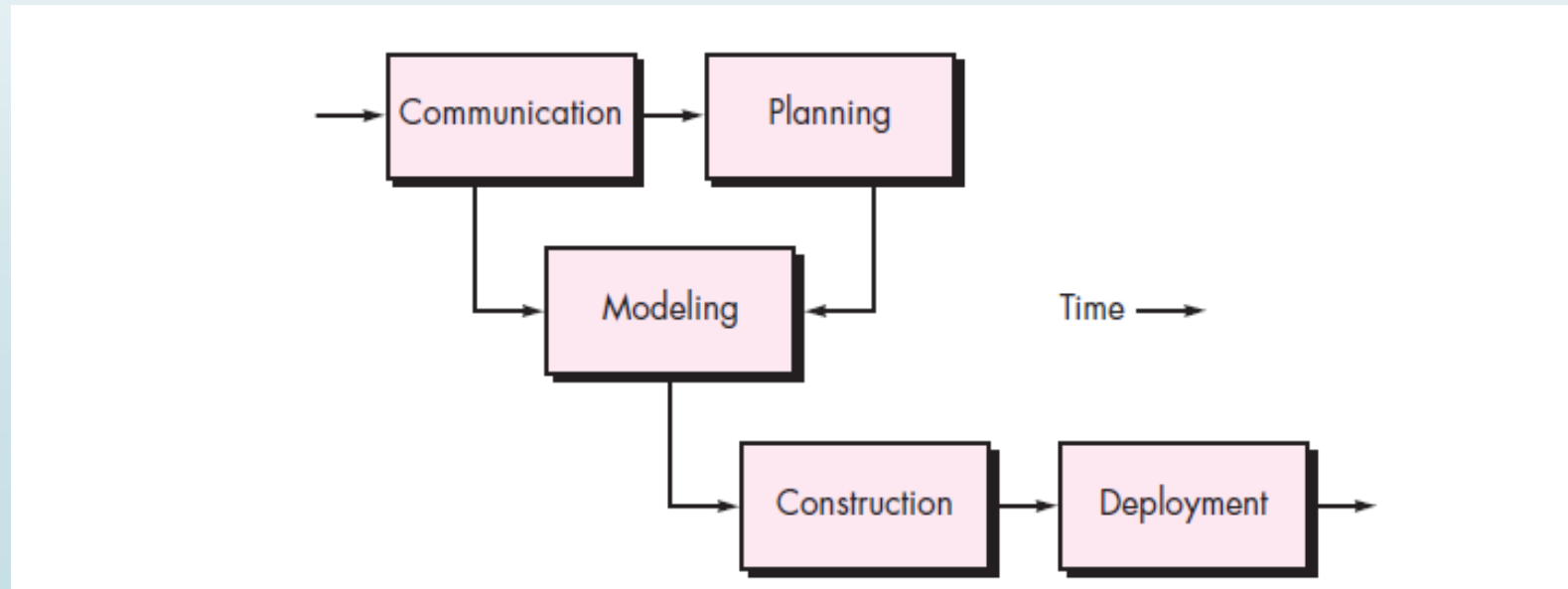
An evolutionary process flow executes the activities in a “circular” manner. Each circuit through the five activities leads to a more complete version of the software.





## 4. Parallel Process Flow:

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).





# Thank You