Process Models: Perspective Process Models

Lecture # 4

Lecture by Engr. Sidra

Objectives

- The objectives of this lecture is to
  - Understand the concept of software processes
  - Understand the generic framework activities that are present in every software process
  - Understand how the processes are modelled and what are process patterns
  - Understand perspective process models and their strength and weaknesses

Software Process

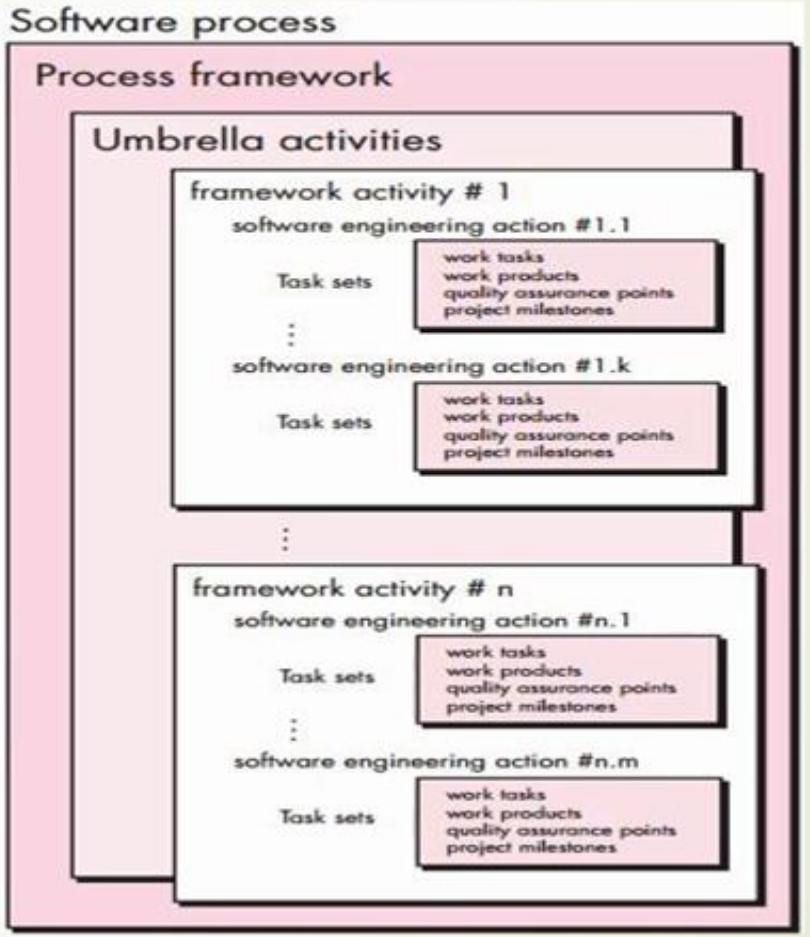
- A process is a collection of activities, actions and tasks that are performed when some work product is to be created.
- It is a adaptable approach that enables people to pick and choose appropriate set of work actions and tasks.

Generic Process Framework Activities

Communication	• Involves communication among the customer and other stake holders, encompasses requirement gathering
Planning	• Establishes a plan for software engineering work, address technical tasks, resources, work products and work schedule
Modelling	• Encompasses the creation of models to better understand the requirements and designs
Construction	• Combines code generation and testing to uncover errors
Deployment	• Involves delivery of software to the customer for evolution and feedback

Software Process Framework

- Process Framework Activities
  - Work tasks
  - Work products
  - Milestones and deliverables
  - Q&A checkpoints
- Umbrella Activities



Software Process Framework

- **Defining a Framework Activity**
  - What actions are appropriate for a framework activity, given
  - the nature of the problem to be solved,
  - the characteristics of the people doing the work,
  - and the stakeholders who are sponsoring the project.

Software Process Framework

- **Identifying a Task set**
  - A task set defines the actual work to be done to accomplish the objectives of a software engineering action.
  - Different projects demand different task sets.
  - The software team chooses
    - A list of the task to be accomplished
    - A list of the work products to be produced
    - A list of the quality assurance filters to be applied

  based on problem and project characteristics.

Software Process Framework

- **Process pattern**
  - It describes a process-related problem that is encountered during software engineering work,
  - Identifies the environment in which the problem has been encountered, and
  - Suggests one or more proven solutions to the problem.
- Stated in more general terms,
  - A process pattern provides you with a template [Amb98]
  - a consistent method for describing problem solutions within the context of the software process.

Software Process Framework

- **Process Pattern Types**
  - Stage patterns—defines a problem associated with a framework activity for the process.
  - Task patterns—defines a problem associated with a software engineering action or work task and relevant to successful software engineering practice
  - Phase patterns—define the sequence of framework activities that occur with the process, even when the overall flow of activities is iterative in nature.
- Once process patterns have been developed, they can be reused for the definition of process variants

Software Process Framework
Proposed Process Pattern Template

- **Pattern Name**: The pattern is given a meaningful name describing it within the context of the software process (e.g., TechnicalReviews).
- **Forces:** The environment in which the pattern is encountered and the issues that make the problem visible and may affect its solution.
- **Type:** The pattern type is specified.
- **Initial context:** Describes the conditions under which the pattern applies.

Software Process Framework
Proposed Process Pattern Template

- **Problem:** The specific problem to be solved by the pattern.
- **Solution:** Describes how to implement the pattern successfully.
- **Resulting Context:** Describes the conditions that will result once the pattern has been successfully implemented.
- **Related Patterns:** Provide a list of all process patterns that are directly related to this one.
- **Known Uses and Examples:** Indicate the specific instances in which the pattern is applicable.

Software Process Framework
Proposed Process Pattern Template

INFO

An Example Process Pattern

The following abbreviated process pattern describes an approach that may be applicable when stakeholders have a general idea of what must be done but are unsure of specific software requirements.

Pattern name. **RequirementsUnclear**

Intent. This pattern describes an approach for building a model (a prototype) that can be assessed iteratively by stakeholders in an effort to identify or solidify software requirements.

Type. Phase pattern.

Initial context. The following conditions must be met prior to the initiation of this pattern: (1) stakeholders have been identified; (2) a mode of communication between stakeholders and the software team has been established; (3) the overriding software problem to be solved has been identified by stakeholders; (4) an initial understanding of project scope, basic business requirements, and project constraints has been developed.

Problem. Requirements are hazy or nonexistent, yet there is clear recognition that there is a problem to be solved, and the problem must be addressed with a software solution. Stakeholders are unsure of what they want; that is, they cannot describe software requirements in any detail.

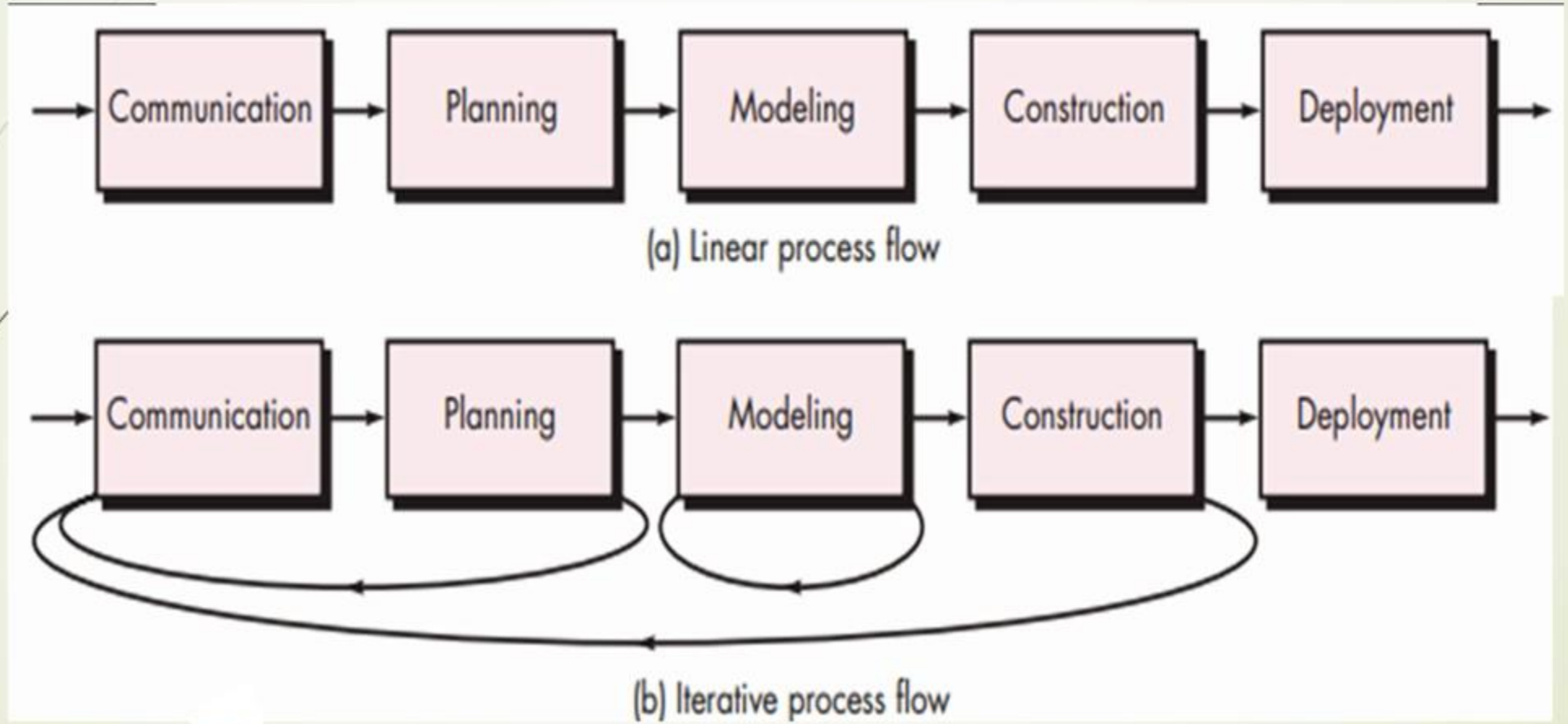
Solution. A description of the prototyping process would be presented here and is described later in Section 2.3.3.

Resulting context. A software prototype that identifies basic requirements (e.g., modes of interaction, computational features, processing functions) is approved by stakeholders. Following this, (1) the prototype may evolve through a series of increments to become the production software or (2) the prototype may be discarded and the production software built using some other process pattern.

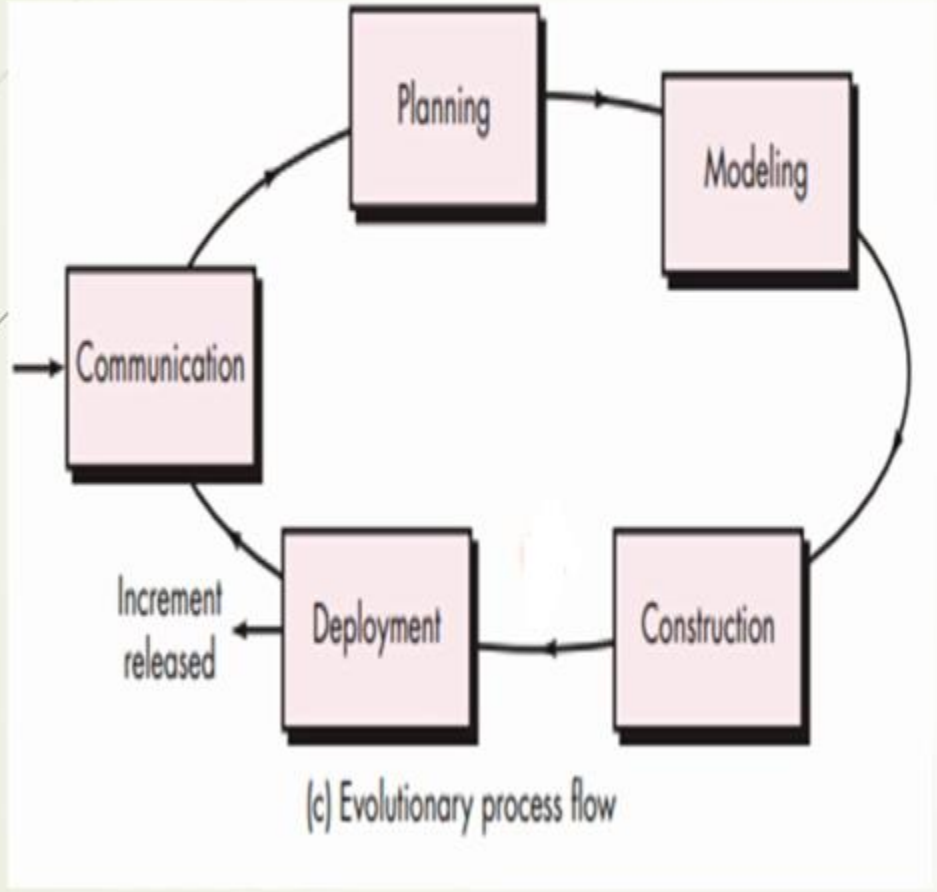
Related patterns. The following patterns are related to this pattern: **CustomerCommunication, IterativeDesign, IterativeDevelopment, CustomerAssessment, RequirementExtraction.**

Known uses and examples. Prototyping is recommended when requirements are uncertain.

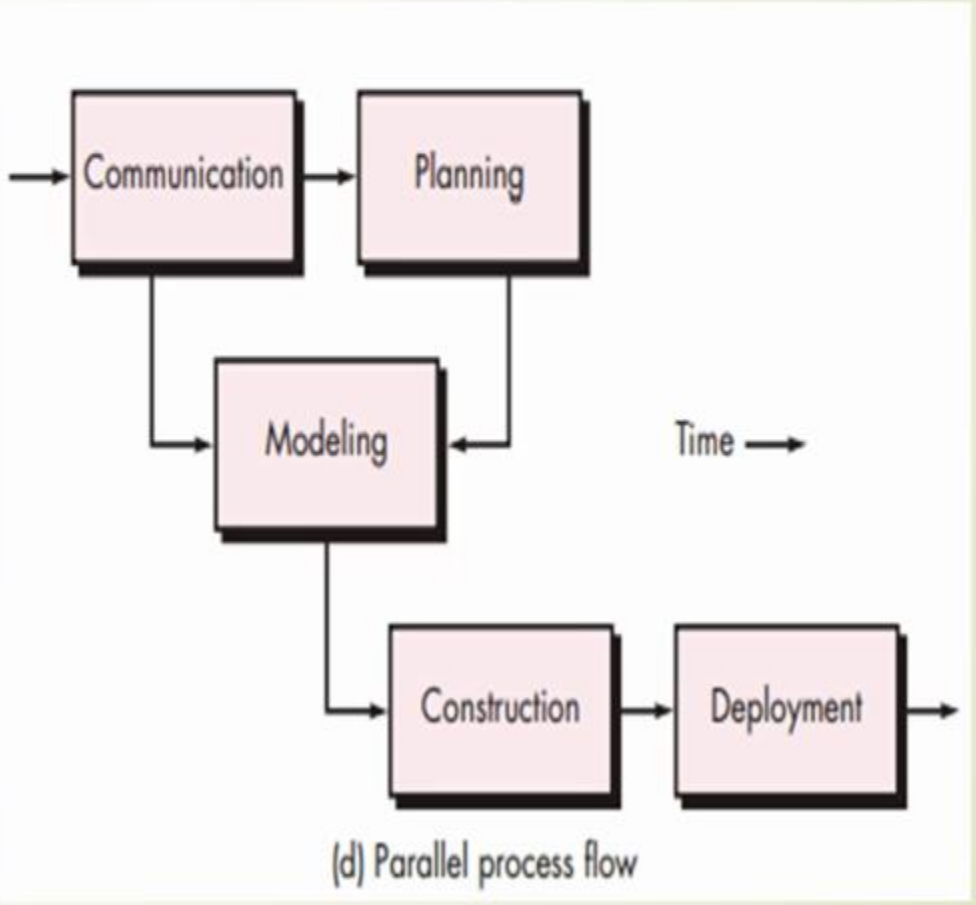
Process Flow



Process Flow



(c) Evolutionary process flow



(d) Parallel process flow

How do process models differ from one another?

- Overall flow of activities, actions, and tasks and the interdependencies among them
- Degree to which actions and tasks are defined within each framework activity
- Degree to which work products are identified and required
- Manner in which quality assurance activities are applied
- Manner in which project tracking and control activities are applied
- Overall degree of detail and rigor with which the process is described
- Degree to which the customer and other stakeholders are involved with the project
- Level of autonomy given to the software team
- Degree to which team organization and roles are prescribed