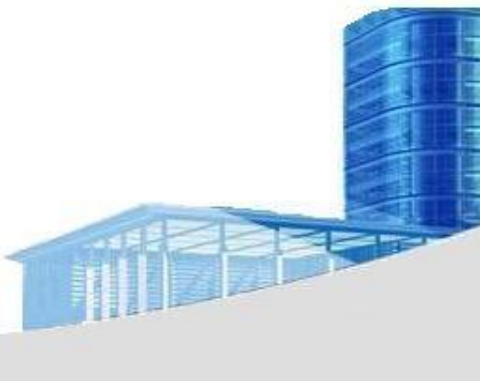
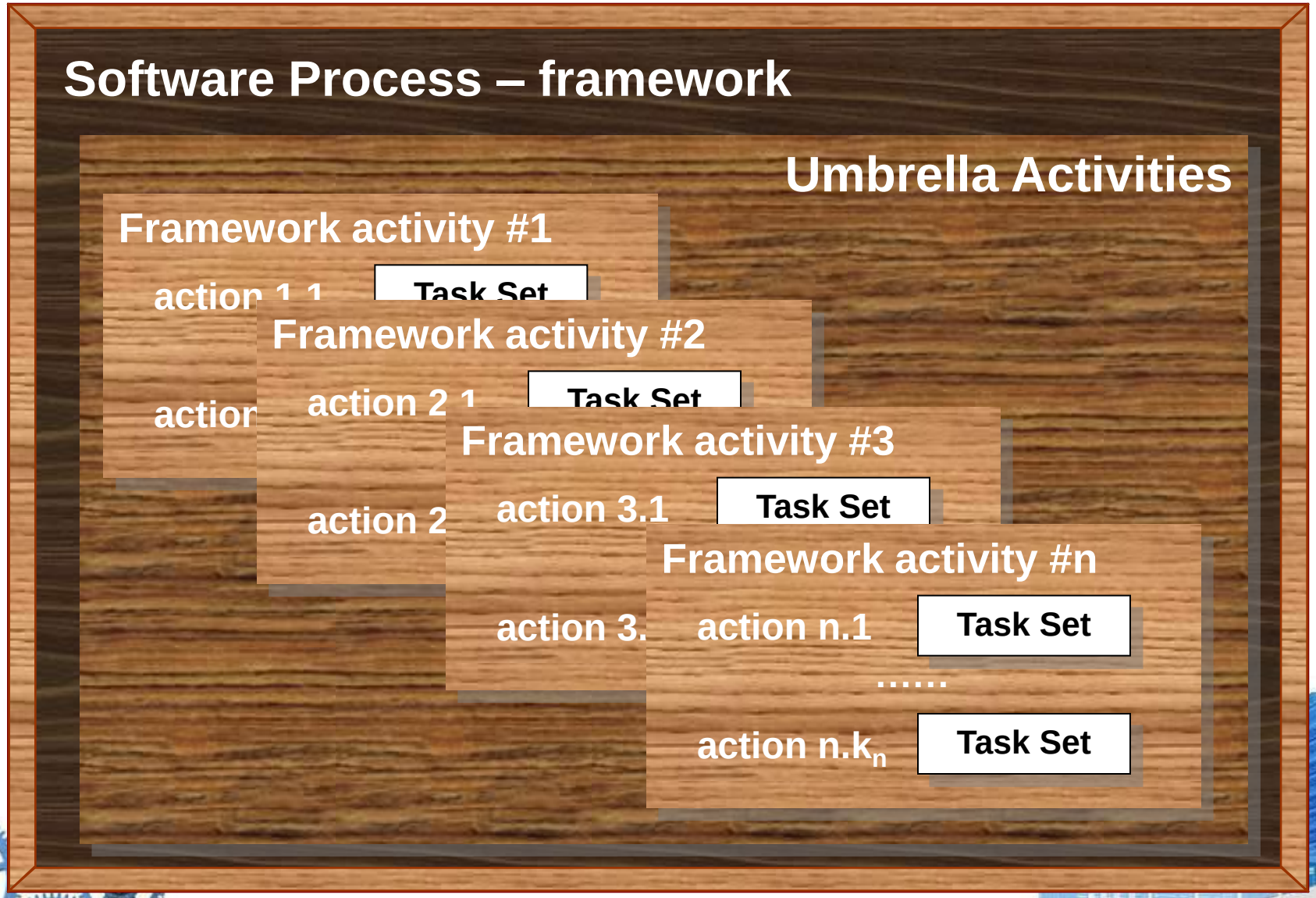
Ch.3 Software Process Structure

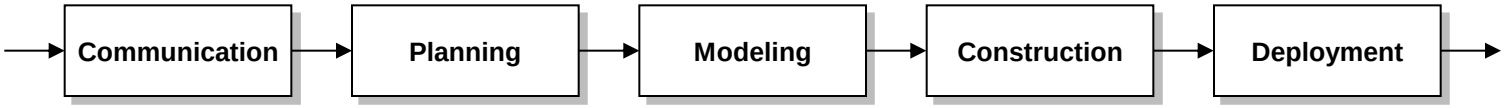


3.1 A Generic Process Model

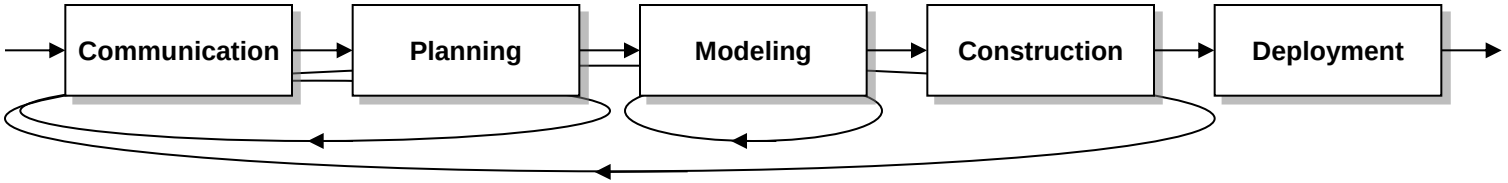


3.1 A Generic Process Model

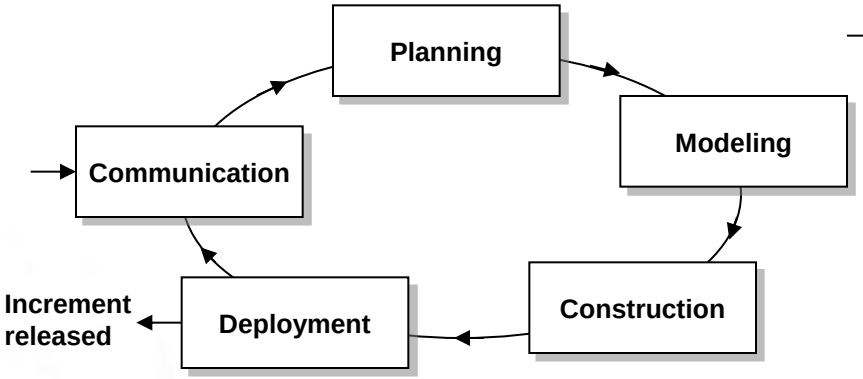
- **Process flow**



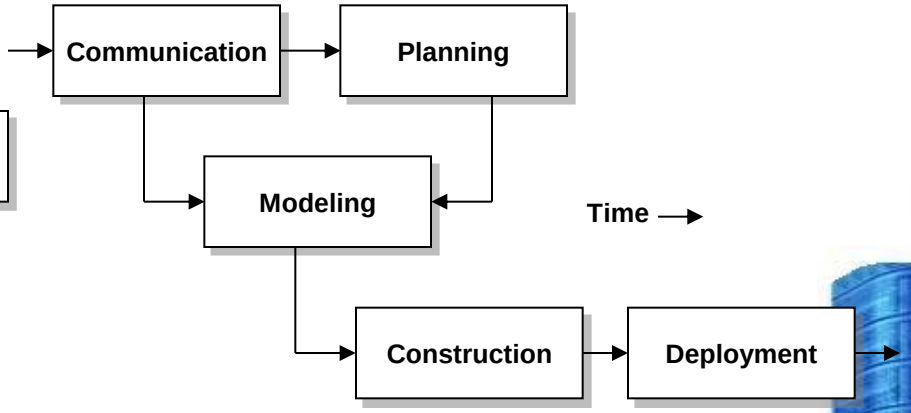
(a) Linear process flow



(b) Iterative process flow



(c) Evolutionary process flow

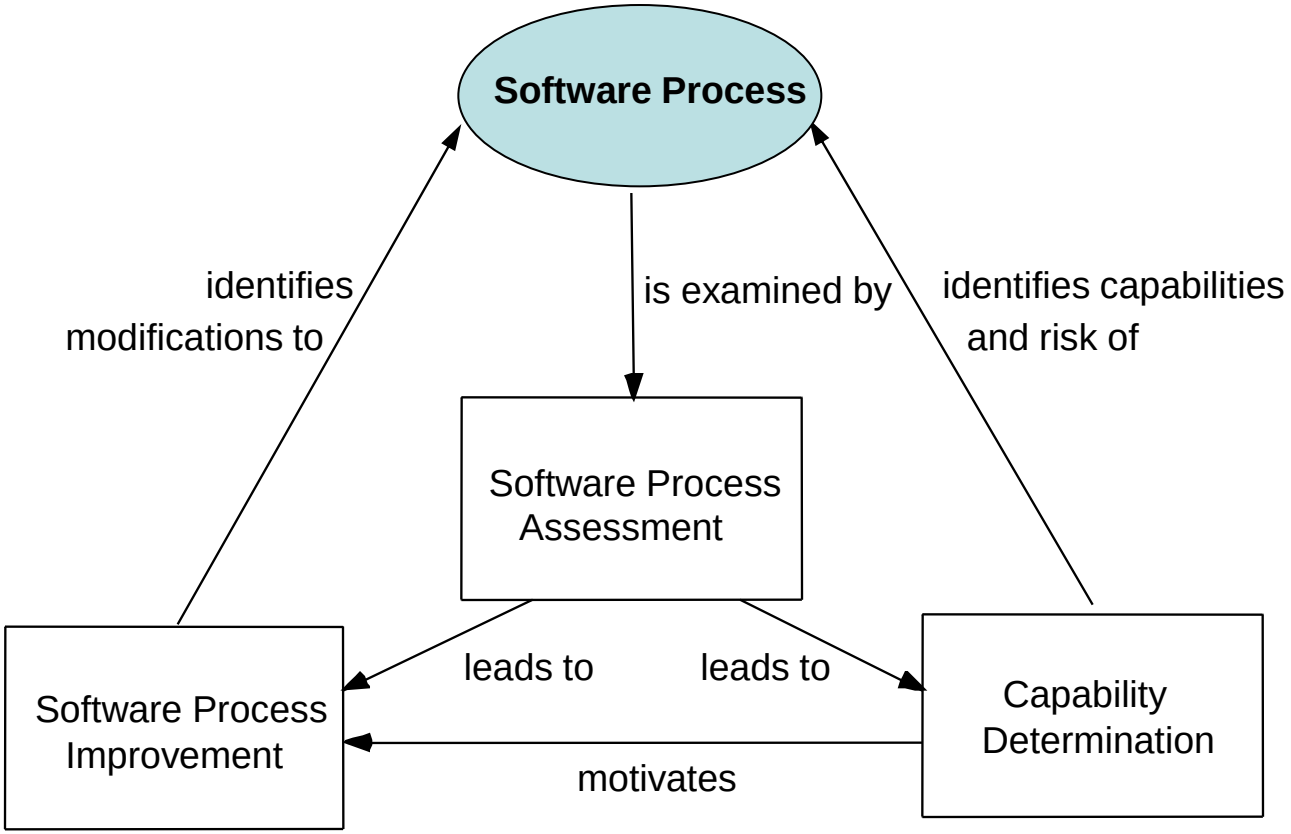


(d) Parallel process flow

3.4 Process Patterns

- **Process patterns** define a set of activities, actions, work tasks, work products and/or related behaviors
- A **template** is used to define a pattern
- **Generic software pattern elements**
  - –Meaningful **pattern name**
  - –**Intent** (objective of pattern)
  - –**Type**
    - Task pattern (defines engineering action or work task)
    - Stage pattern (defines framework activity for the process)
    - Phase pattern (defines sequence or flow of framework activities that occur within process)
  - –**Initial context** (describes conditions that must be present prior to using pattern)
  - –**Solution** (describes how to implement pattern correctly)
  - –**Resulting context** (describes conditions that result when pattern has been implemented successfully)
  - –**Related patterns** (links to patterns directly related to this one)
  - –**Known uses/examples** (instances in which pattern is applicable)

3.5 Process Assessment



- SCAMPI
- SPICE (ISO/IEC15504)
- CBA IPI
- ISO 9001:2000 for Software

The Capability Maturity Model Integration

— by Software Engineering Institute (SEI) of Carnegie Mellon University (CMU)

- **Level 0: Incomplete** (process is not performed or does not achieve all goals defined for this level)
- **Level 1: Performed** (work tasks required to produce required work products are being conducted)
- **Level 2: Managed** (people doing work have access to adequate resources to get job done, stakeholders are actively involved, work tasks and products are monitored, reviewed, and evaluated for conformance to process description)
- **Level 3: Defined** (management and engineering processes documented, standardized, and integrated into organization-wide software process)
- **Level 4: Quantitatively Managed** (software process and products are quantitatively understood and controlled using detailed measures)
- **Level 5: Optimizing** (continuous process improvement is enabled by quantitative feedback from the process and testing innovative ideas)