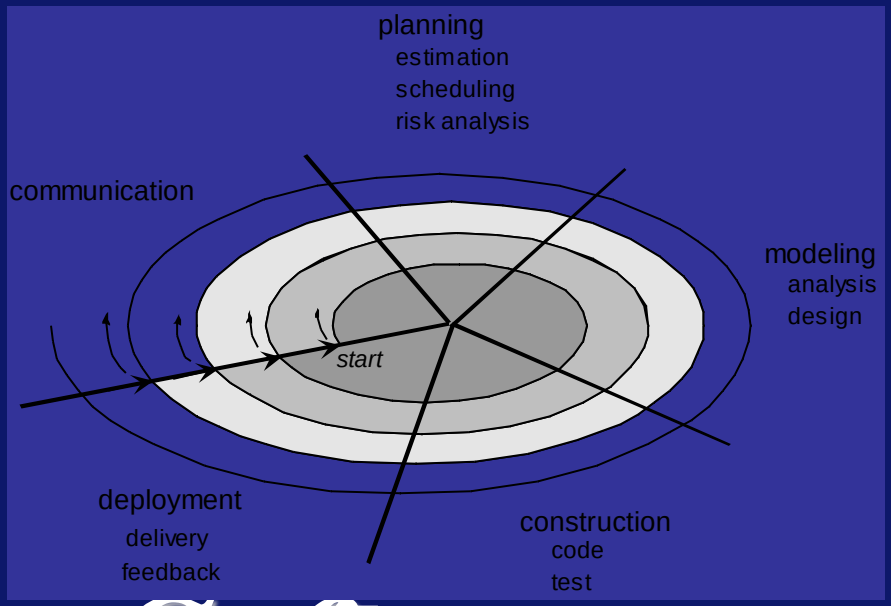


Software Engineering

Chapter 3
Prescriptive Process Models

Moon kun Lee
Division of Electronics & Information Engineering
Chonbuk National University

Prescriptive vs. Agile

- Prescriptive [Chapter 3]
  - Order and project consistency are dominant issues
- Agile [Chapter 4]
  - Self-organization, collaboration, communication, and adaptability dominate the process philosophy.

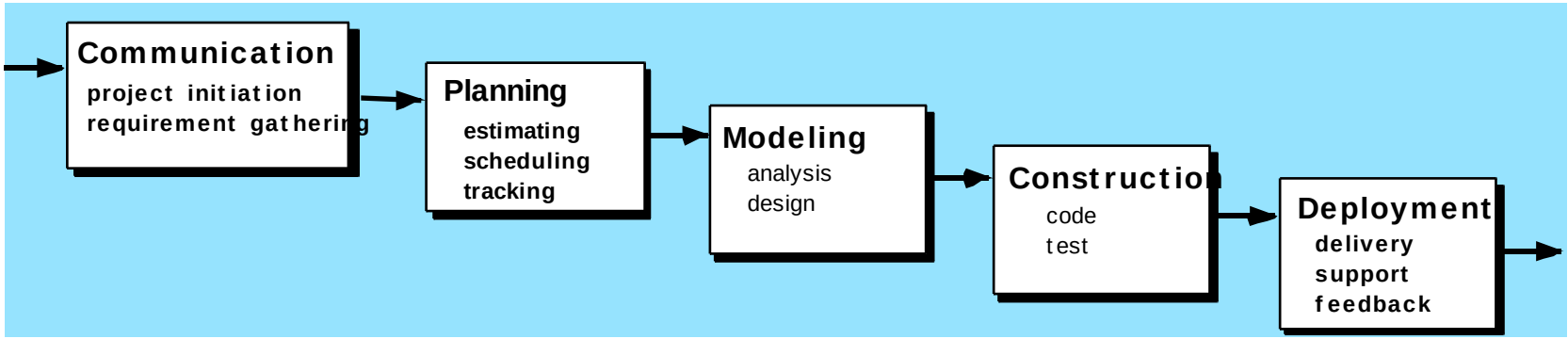
Prescriptive Models

- Prescriptive process models advocate an orderly approach to software engineering

That leads to a few questions …

- If prescriptive process models strive for structure and order, are they inappropriate for a software world that thrives on change?
- Yet, if we reject traditional process models (and the order they imply) and replace them with something less structured, do we make it impossible to achieve coordination and coherence in software work?

The Waterfall Model

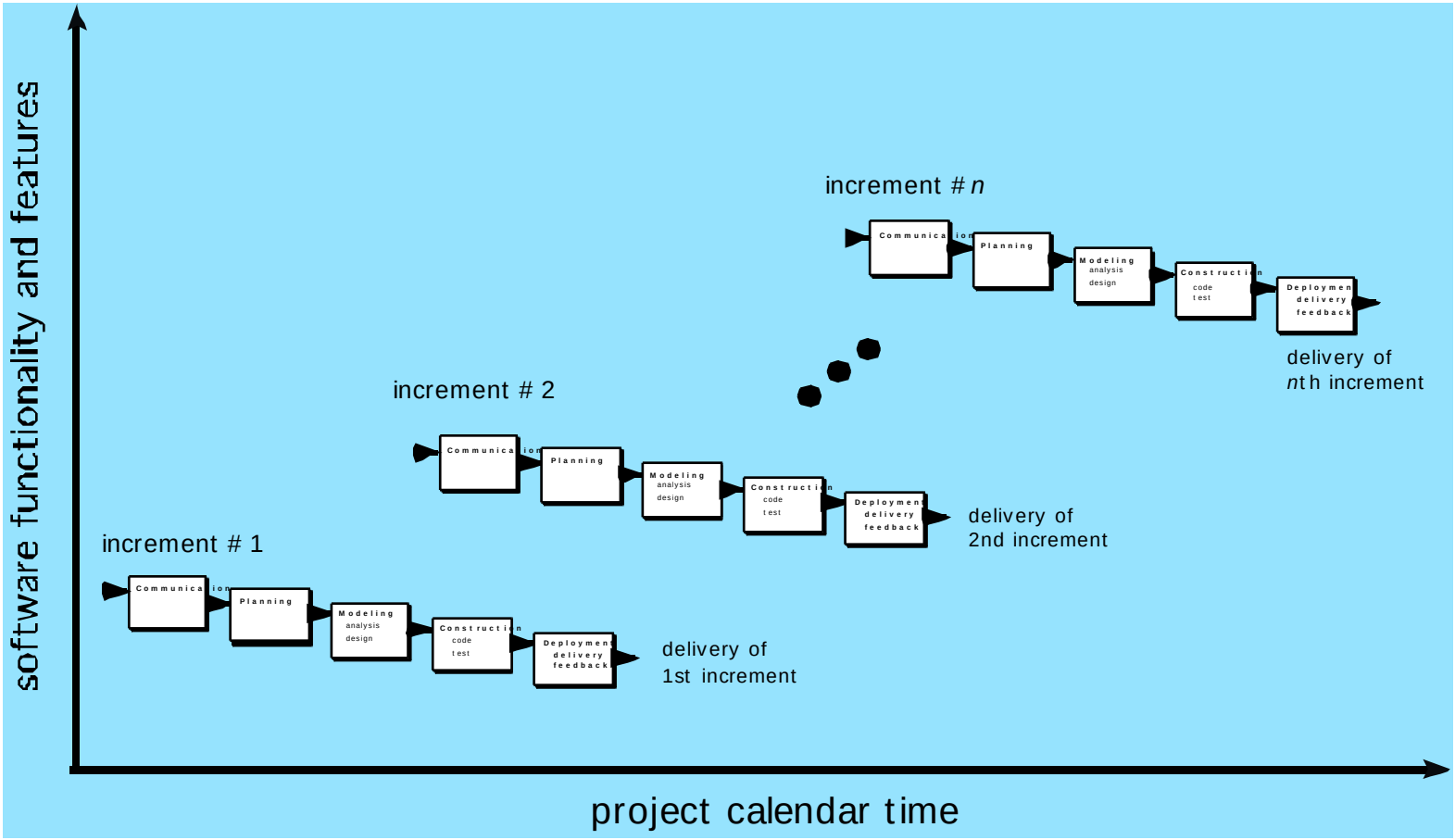


- The classic life cycle.
- The oldest paradigm.
- Suggest a systematic, sequential approach to software development that begins with customer specification of requirements and progresses through planning, modeling, construction, and deployment, culminating in on-going support of the complete software.

Problems

- Real projects rarely follow the sequential flow that the model proposes.
- It is often difficult for the customer to state all requirements explicitly.
- The customer must have patience.

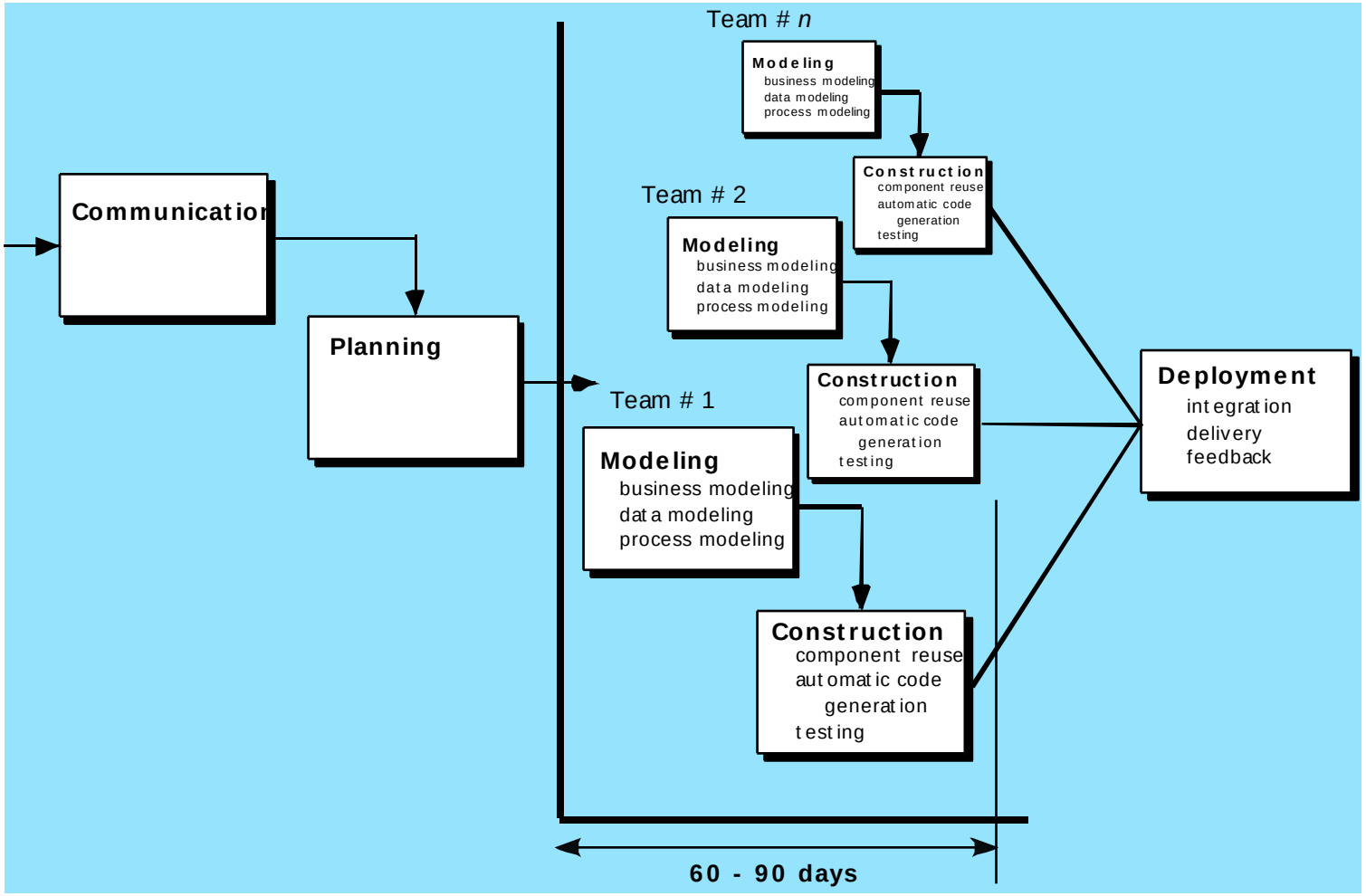
The Incremental Model



- Characteristics
  - Incremental
  - Iterative
- Pro
  - Useful when staffing is unavailable for a complete implementation by the business deadline.
  - Man-power management
- Problems
  - Technical risks
  - Time-consuming

The RAD Model

- Rapid Application Development



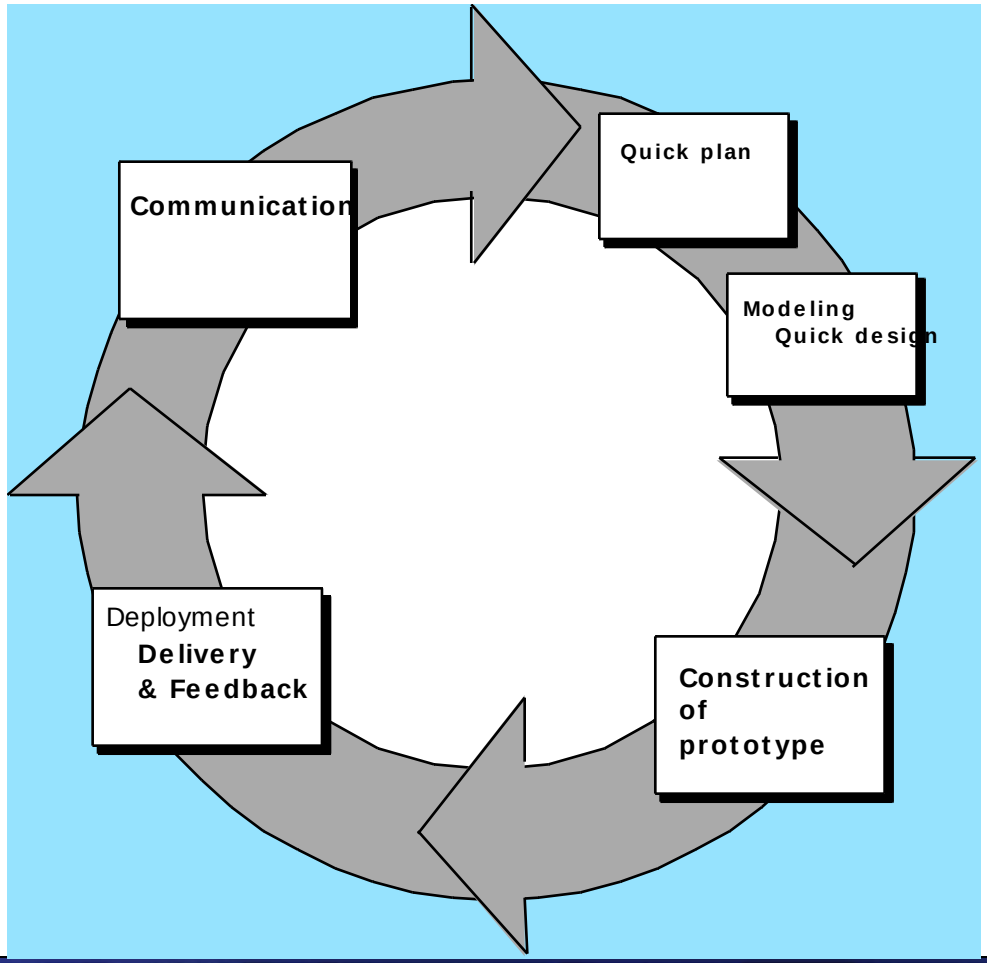
- An incremental SW process model that emphasizes a short development cycle.
- A high-speed adaptation of the waterfall model, in which rapid development is achieved by using a component based construction approach.
- Requirements must be well understood and project scope be constrained.
- Can be mapped into the generic framework activities.
- Applications must be modularized.
- Drawbacks
  - Sufficient human resources
  - Rapid-fire activities between developers & customers
  - Modularization
  - High performance & tuning interfaces
  - High technical risks

Evolutionary Models

- Adaptation to the nature of SW, i.e., evolution.
- Iterative
- Enables SW engineers to develop increasingly more complete versions of the SW.

Evolutionary Models: Prototyping

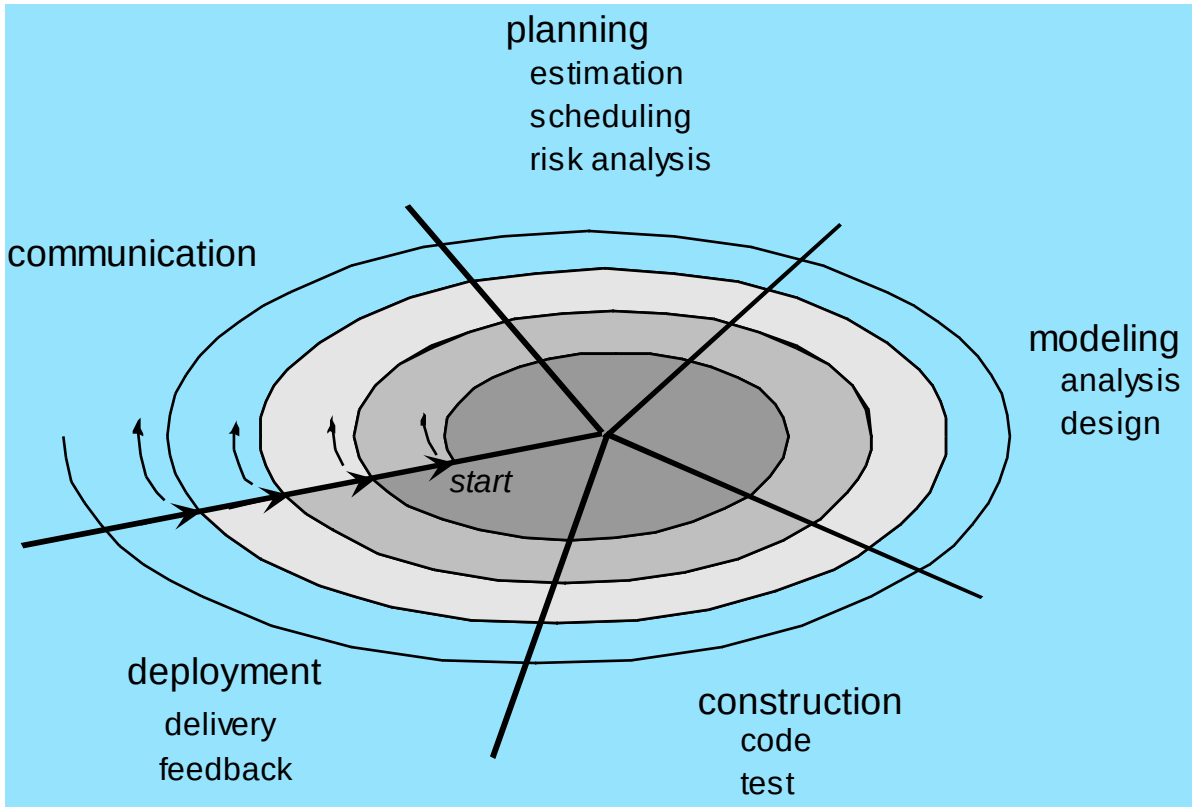
- Not fully certain, not fully ready.



- The quick design focuses on a representation of those aspects of the SW that will be visible to the customer/end-user.
- The quick design leads to the construction of a prototype.
- The prototype is deployed and then evaluated by the customer/user.
- Iteration.
- Problematic:
  - Software quality & long-term maintainability
  - Quick implementation of a prototype: improper OS, PL, etc.

Evolutionary Models: The Spiral

- An model that couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model.

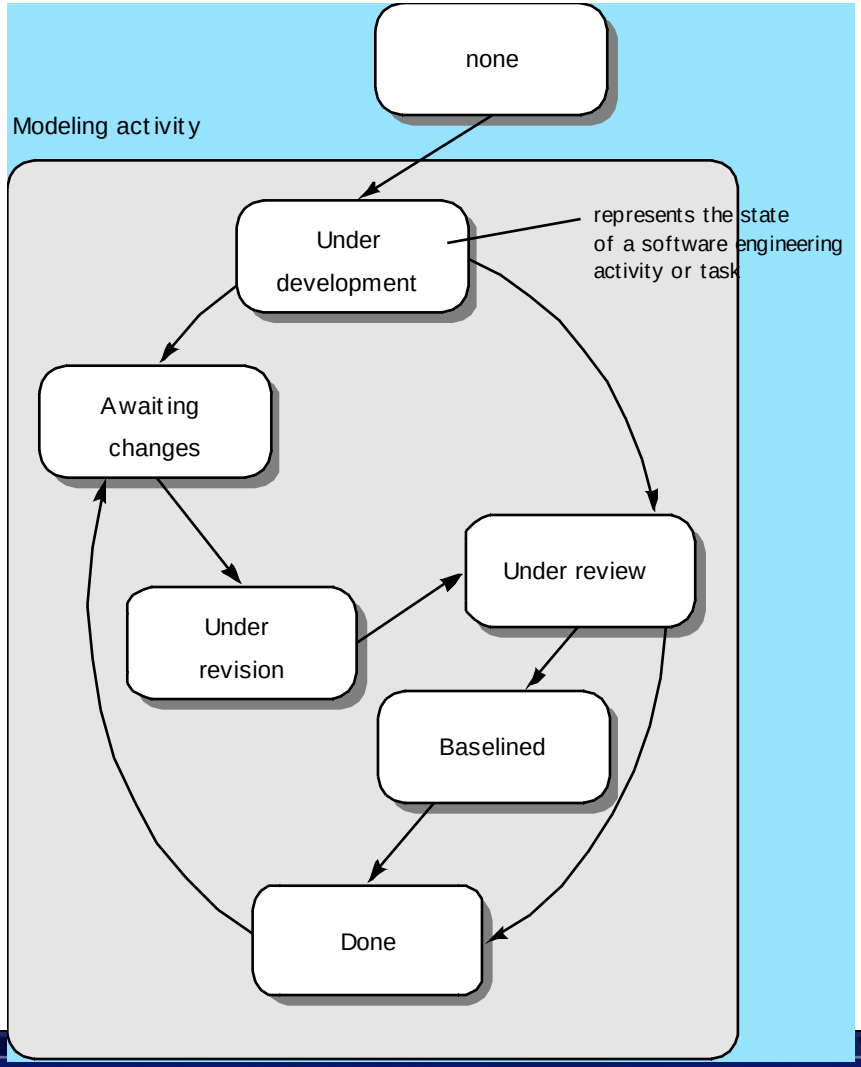


- Divided into a set of framework activities.
- Can be adapted to apply throughout the life of the SW.
- A realistic approach to the development of large-scale systems and SW.
- Drawbacks:
  - Difficult to convince customers that the approach is controllable.
  - Demands considerable risk assessment expertise and relies on this expertise for success.

Evolutionary Models: Concurrent

- Represented as a series of activities, actions & tasks.

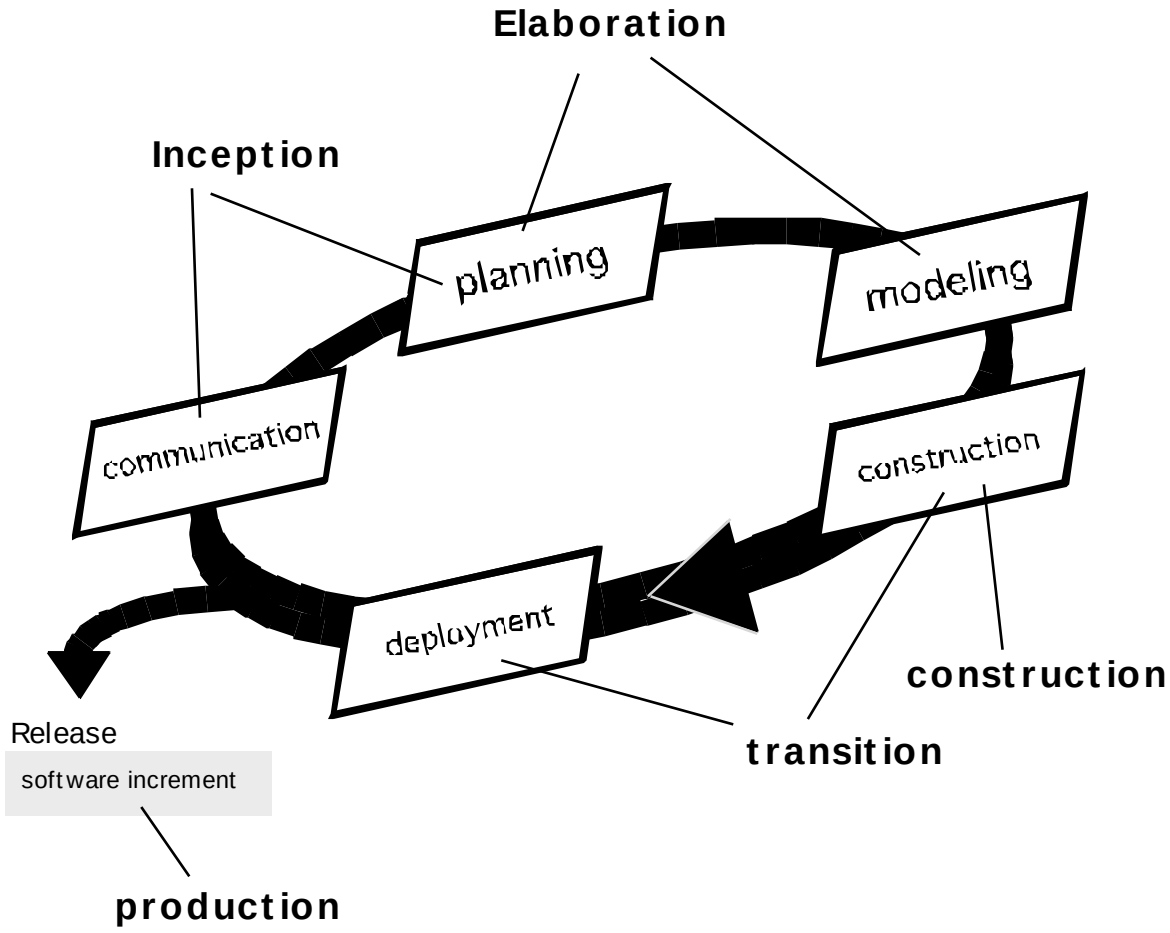
•Prototyping
•Analysis modeling
•Specification
•Design



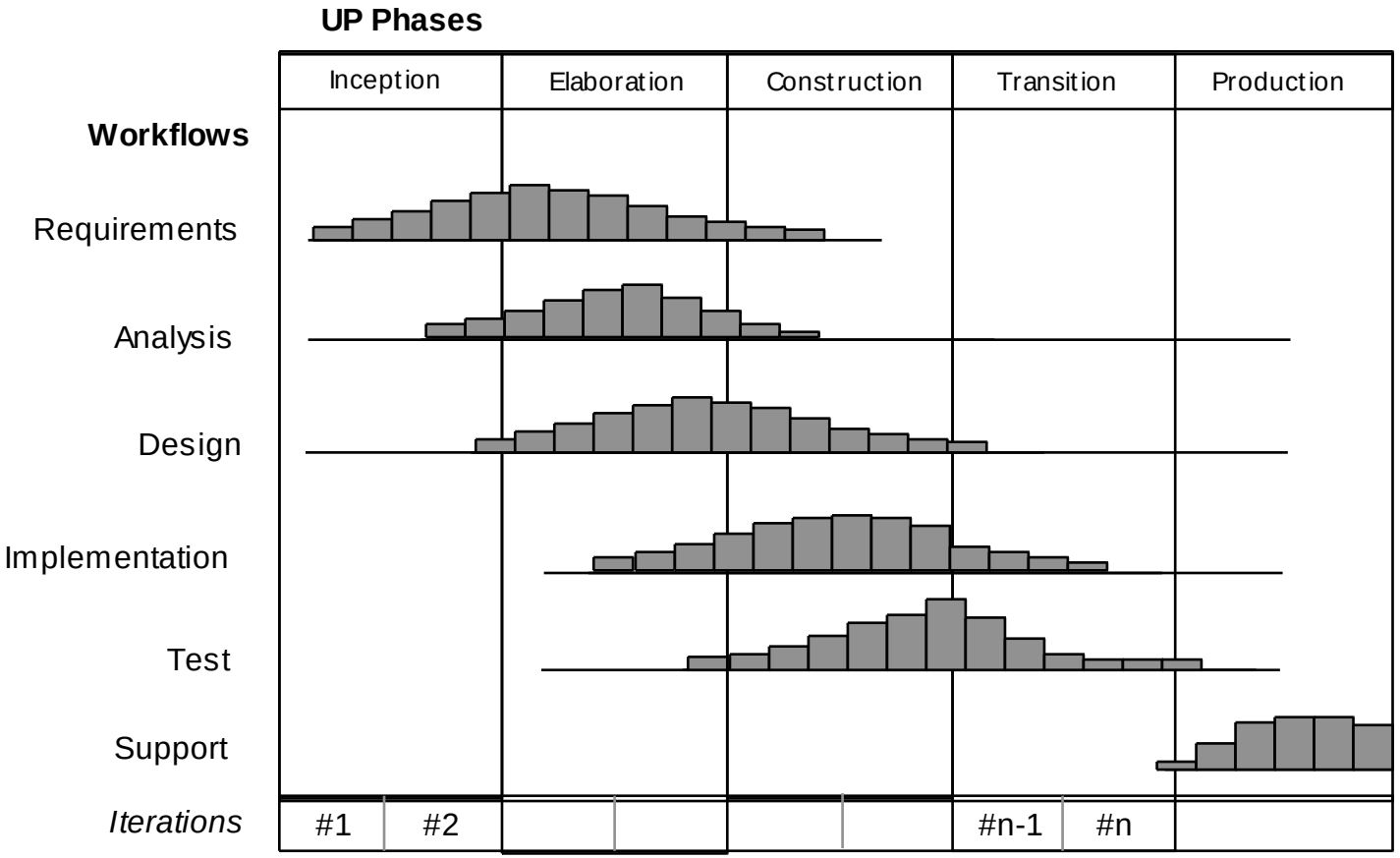
Still Other Process Models

- Component based development—the process to apply when reuse is a development objective
- Formal methods—emphasizes the mathematical specification of requirements
- AOSD(Aspect-Oriented Component Engineering)—provides a process and methodological approach for defining, specifying, designing, and constructing *aspects*
- Unified Process—a "use-case driven, architecture-centric, iterative and incremental" software process closely aligned with the Unified Modeling Language (UML)

The Unified Process (UP)



UP Phases


UP Phases
Inception
Elaboration
Construction
Transition
Production
Workflows
Requirements
Analysis
Design
Implementation
Test
Support
Iterations
#1
#2
#n-1
#n

UP Work Products