Chapter 2

- **Process Models**

Slide Set to accompany

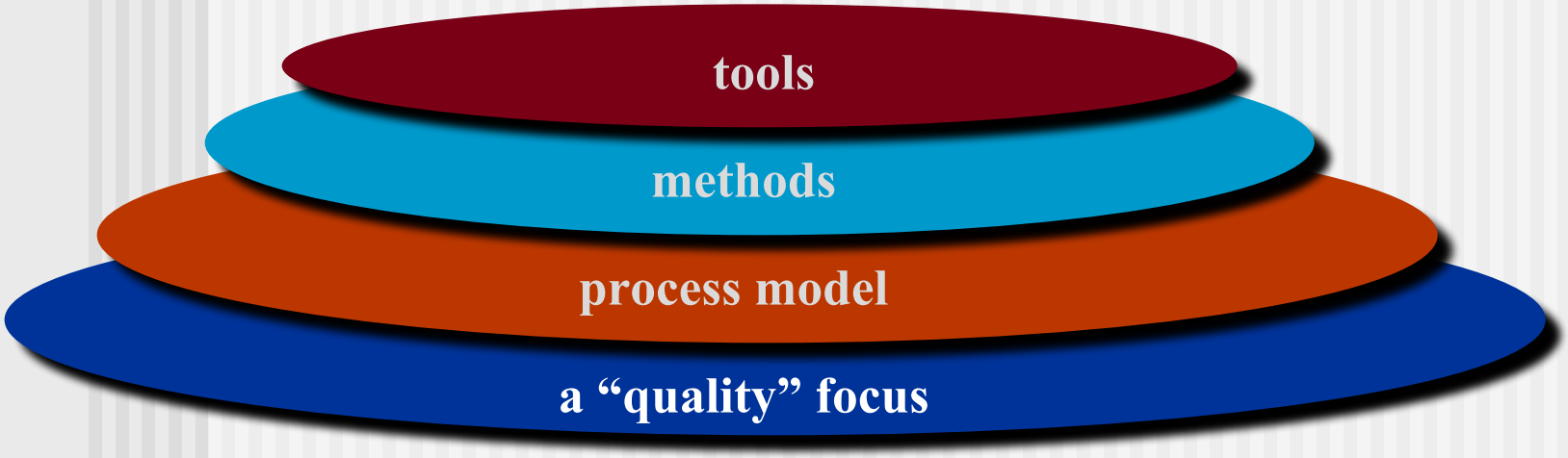
Software Engineering: A Practitioner's Approach, 7/e

by Roger S. Pressman

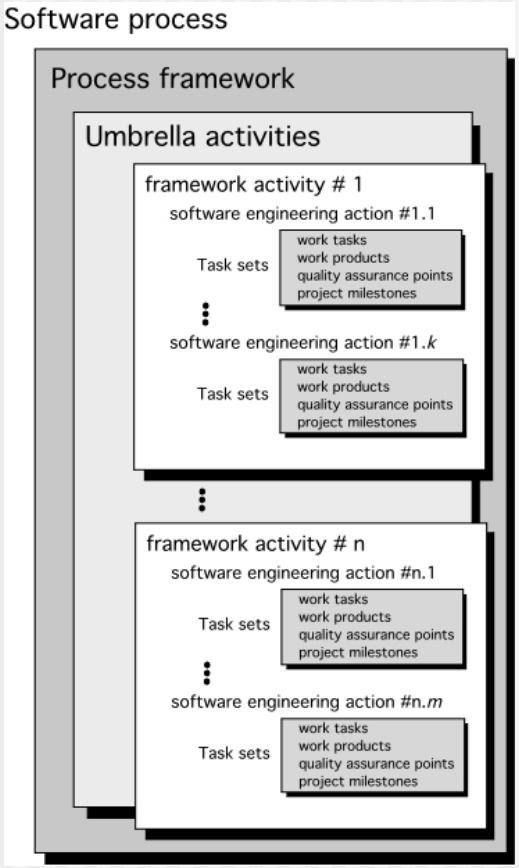
Slide Set - 3

A Layered Technology

Software Engineering



A Generic Process Model



A Process Framework

Process framework

- Framework activities
  - work tasks
  - work products
  - milestones & deliverables
  - QA checkpoints
- Umbrella Activities

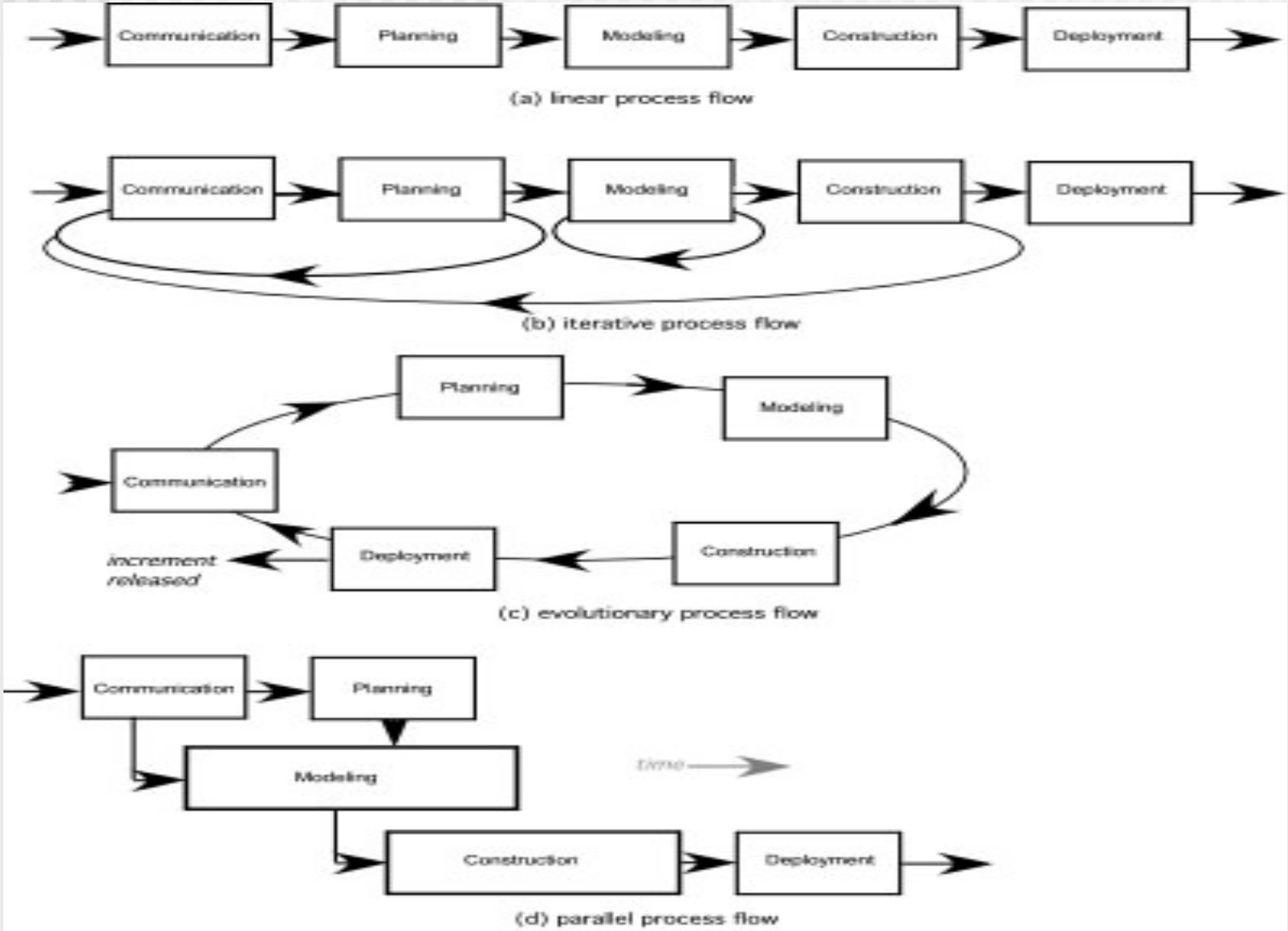
Framework Activities

- Communication
- Planning
- Modeling
  - Analysis of requirements
  - Design
- Construction
  - Code generation
  - Testing
- Deployment

Umbrella Activities

- Software project management
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Work product preparation and production
- Reusability management
- Measurement
- Risk management

Process Flow



Identifying a Task Set

- A task set defines the actual work to be done to accomplish the objectives of a software engineering action.
  - 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

Process Assessment and Improvement

- **Standard CMMI Assessment Method for Process Improvement (SCAMPI)** — provides a five step process assessment model that incorporates five phases: initiating, diagnosing, establishing, acting and learning.
- **CMM-Based Appraisal for Internal Process Improvement (CBA IPI)**—provides a diagnostic technique for assessing the relative maturity of a software organization; uses the SEI CMM as the basis for the assessment [Dun01]
- **SPICE—The SPICE (ISO/IEC15504)** standard defines a set of requirements for software process assessment. The intent of the standard is to assist organizations in developing an objective evaluation of the efficacy of any defined software process. [ISO08]
- **ISO 9001:2000 for Software**—a generic standard that applies to any organization that wants to improve the overall quality of the products, systems, or services that it provides. Therefore, the standard is directly applicable to software organizations and companies. [Ant06]

Software life cycle

Series of identifiable stages that a software product undergoes during its life time:

- Feasibility study
- Requirements analysis and specification,
- Design,
- Coding,
- Testing
- maintenance.

Software Life Cycle

- identifies all the activities required for product development,
- establishes a precedence ordering among the different activities,
- Divides life cycle into phases.

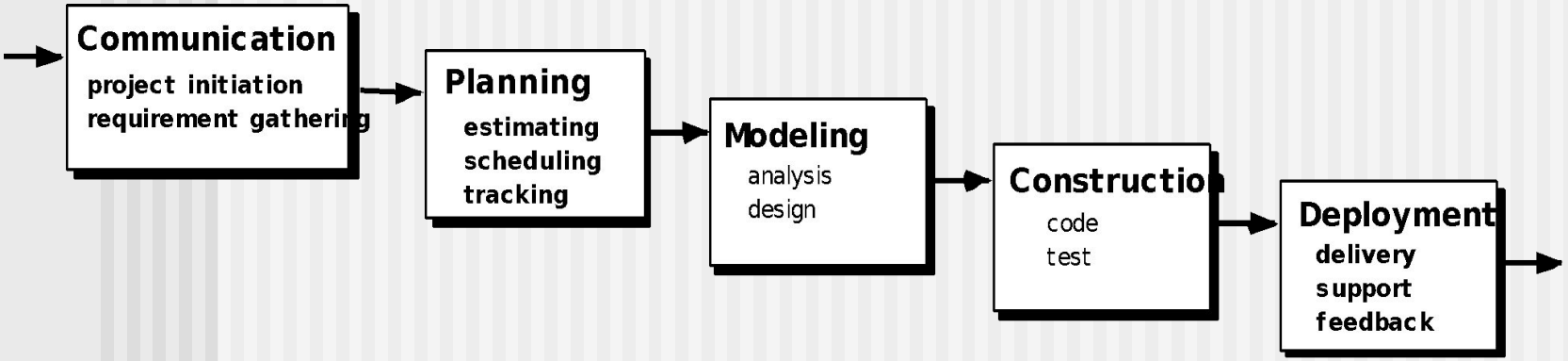
Life Cycle Model is Adhered to,

- the project manager can at any time fairly accurately tell,
  - at which stage (e.g., design, code, test, etc.) of the project is.
- Otherwise, it becomes very difficult to track the progress of the project
  - the project manager would have to depend on the guesses of the team members.

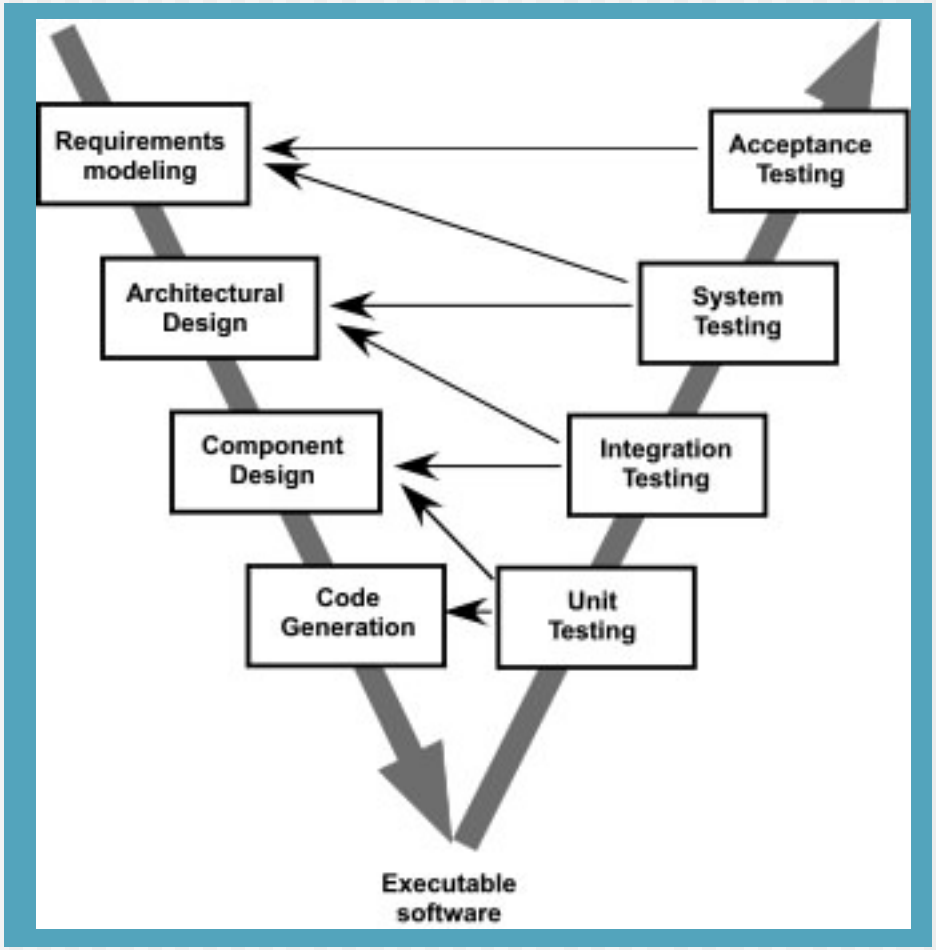
Attention to Specific models

- Classical waterfall model
- Iterative waterfall,
- Evolutionary,
- Prototyping, and
- Spiral model

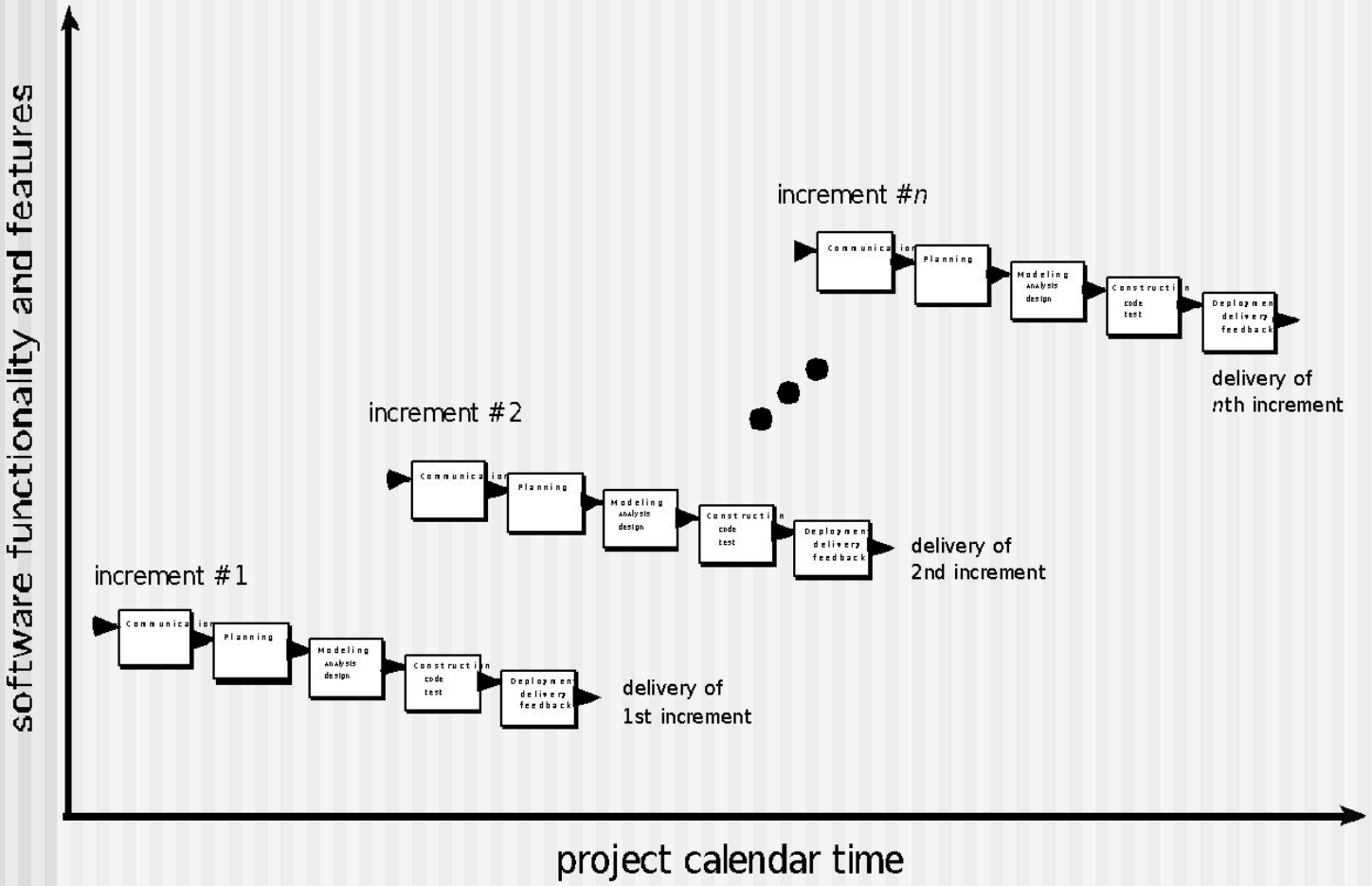
The Waterfall Model



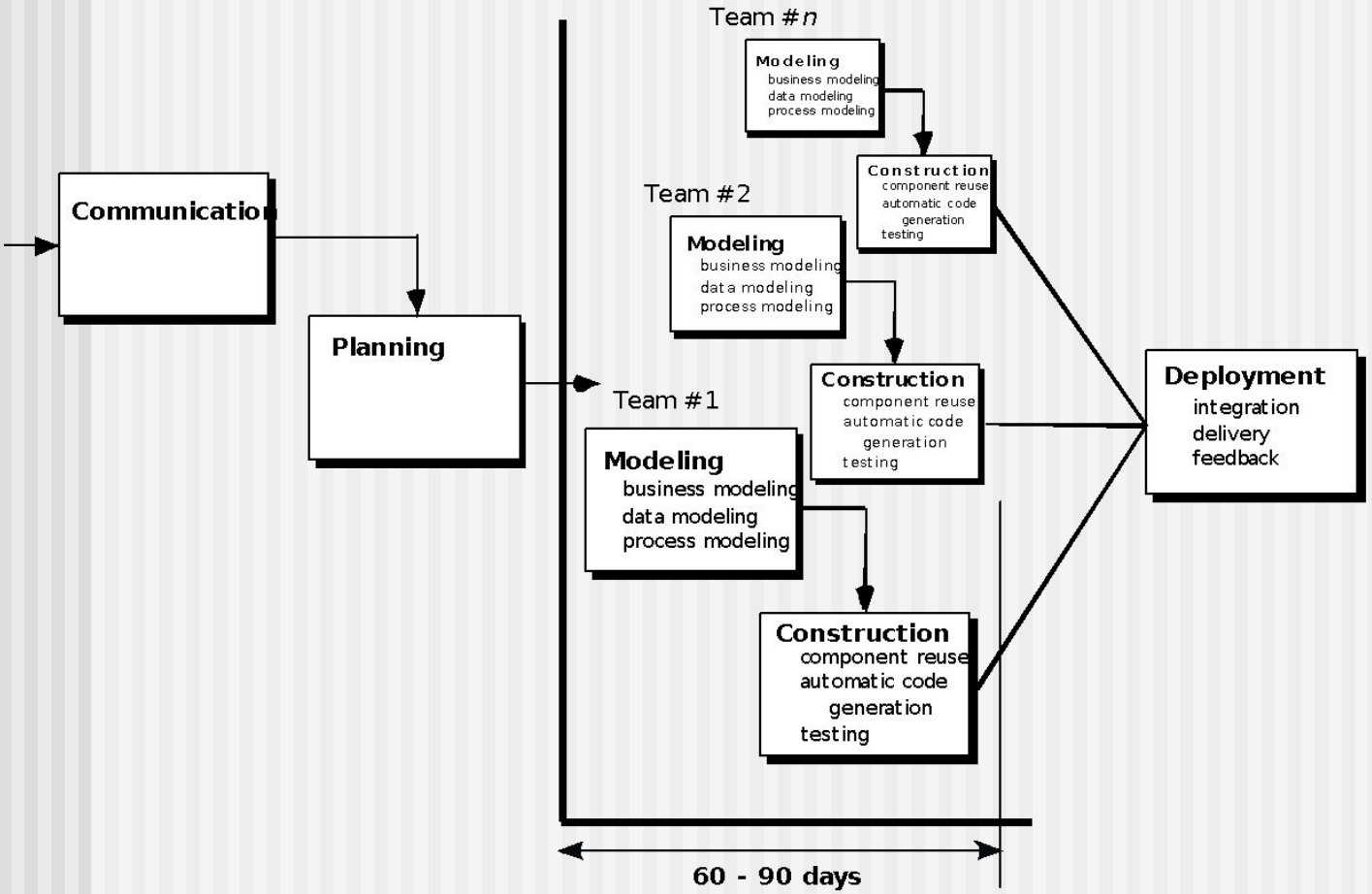
The V-Model


Requirements modeling
Acceptance Testing
Architectural Design
System Testing
Component Design
Integration Testing
Code Generation
Unit Testing
Executable software

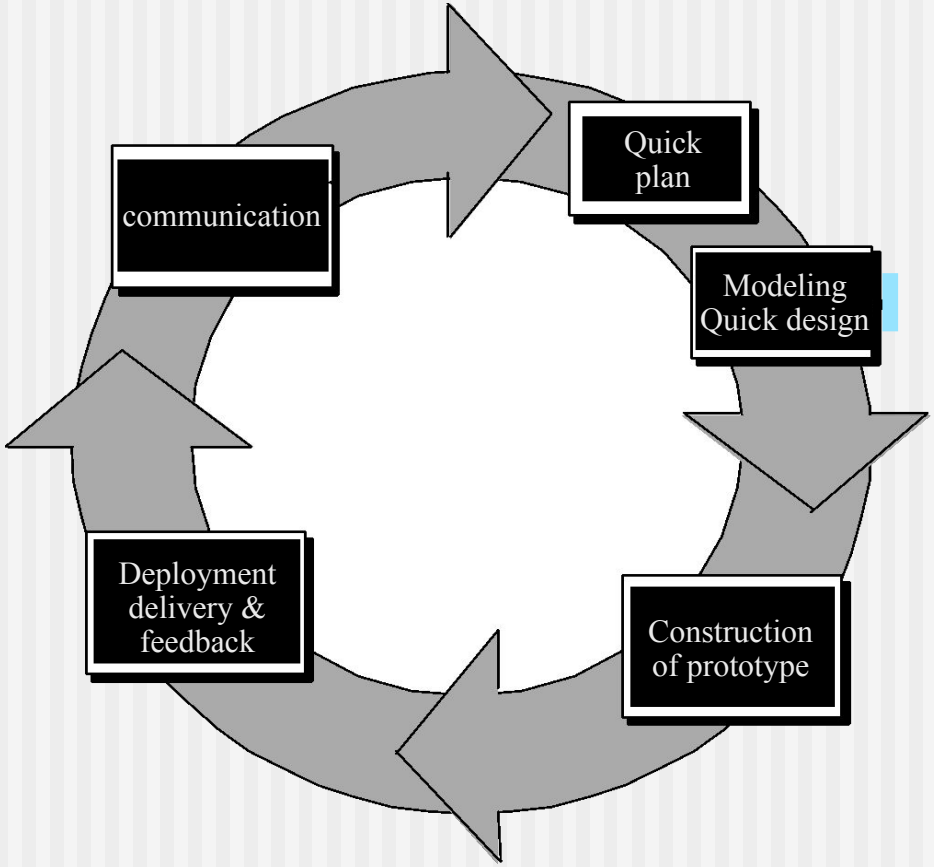
The Incremental Model



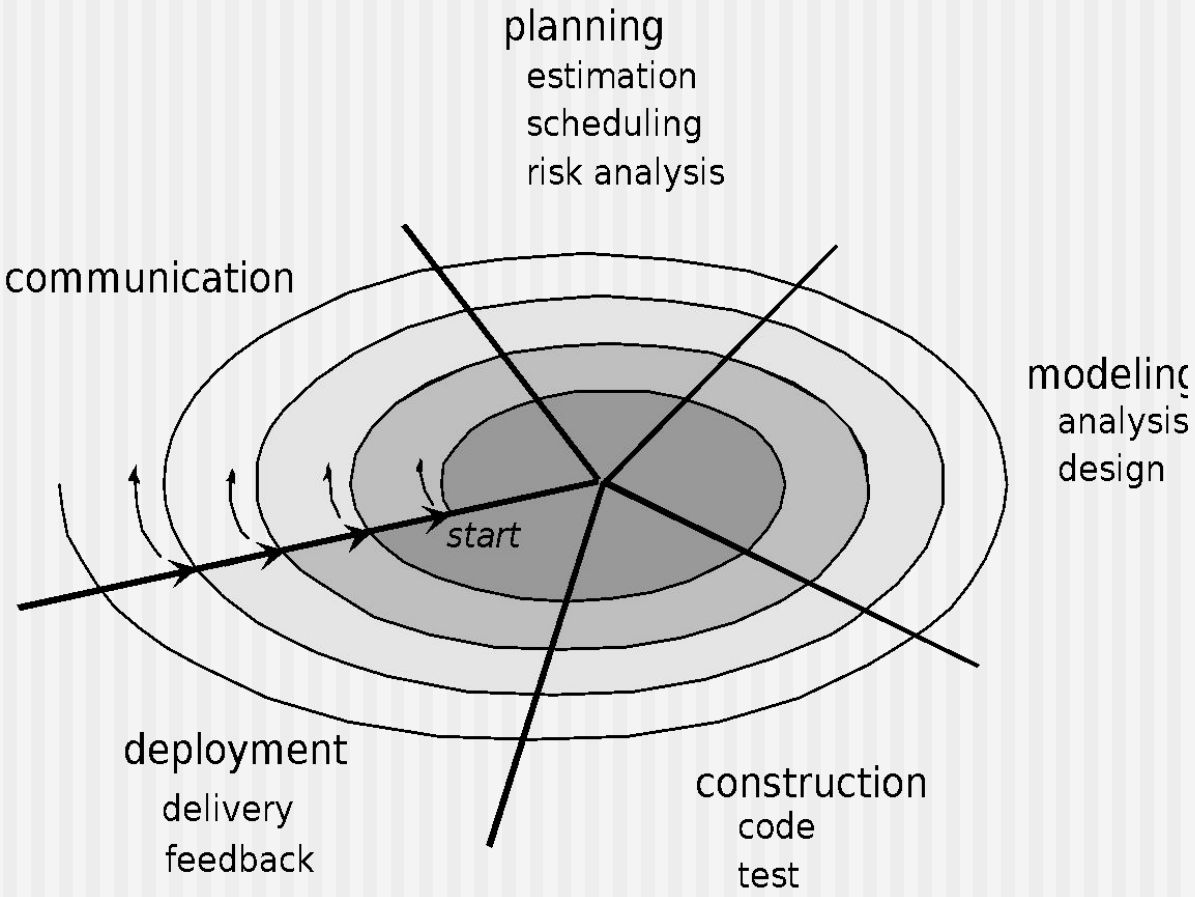
The RAD Model



Evolutionary Models: Prototyping



Evolutionary Models: The Spiral



Iterative vs Incremental Model

Iterative

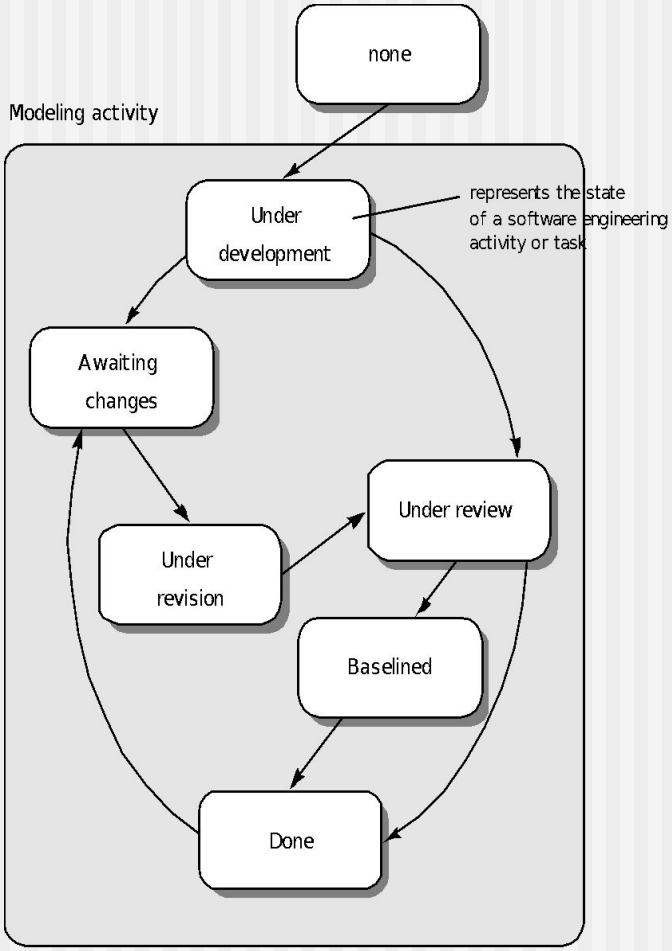
- Develop through repeated cycles
- Start simple, expecting to change
- Used to find the right solution (fail early)
- Used to improve the candidate solution

Incremental

- Develop smaller portions at a time
- Gradually build up functionality
- Allows value to be delivered early

Source: Scrum Guide 6; 7 at slideshare.net

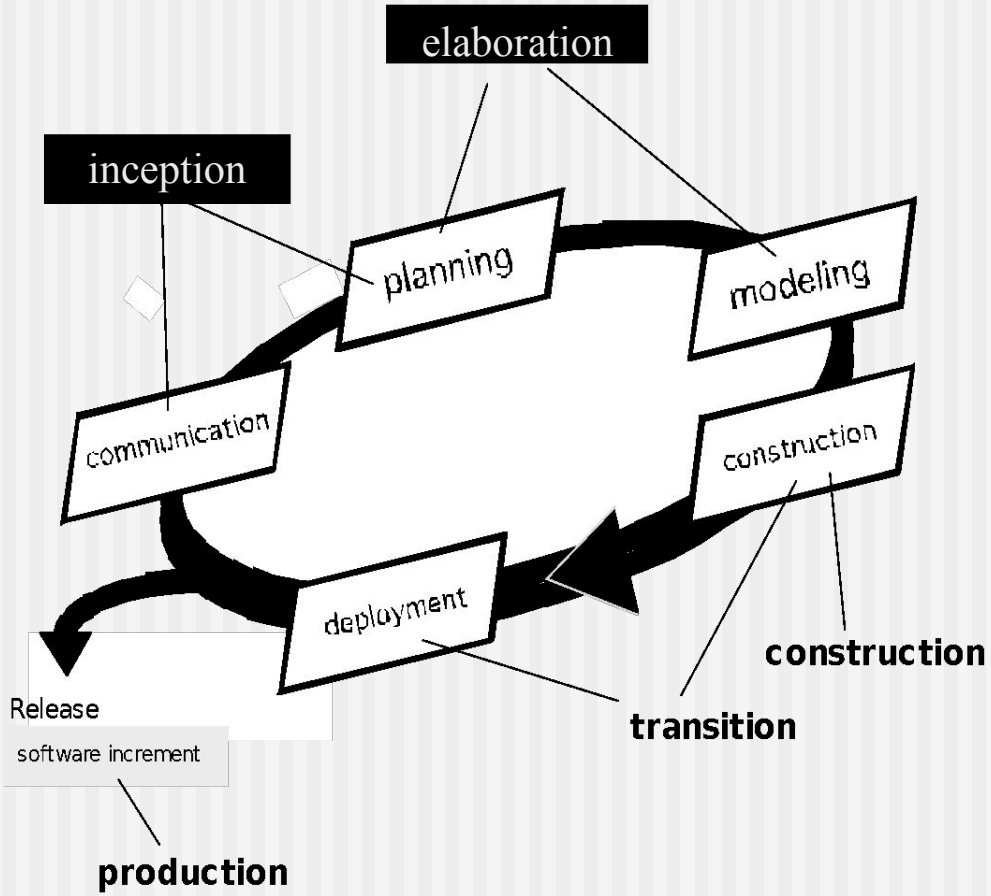
Evolutionary Models: Concurrent



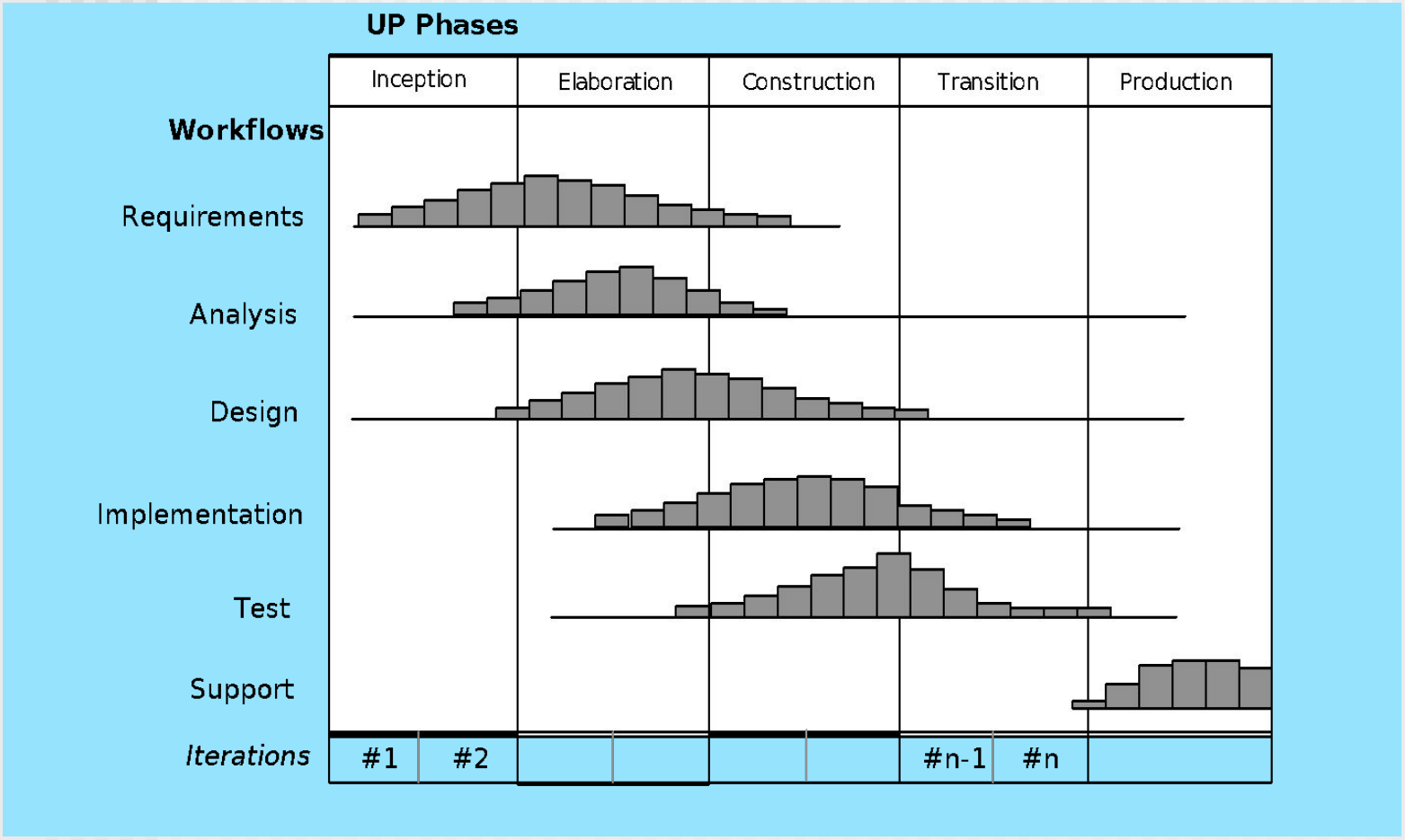
Still Other Process Models

- Component based development—the process to apply when reuse is a development objective
- Formal methods—emphasizes the mathematical specification of requirements
- AOSD—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

Inception phase

Vision document
Initial use-case model
Initial project glossary
Initial business case
Initial risk assessment.
Project plan,
phases and iterations.
Business model,
if necessary.
One or more prototypes

Elaboration phase

Use-case model
Supplementary requirements
including non-functional
Analysis model
Software architecture
Description.
Executable architectural
prototype.
Preliminary design model
Revised risk list
Project plan including
iteration plan
adapted workflows
milestones
technical work products
Preliminary user manual

Construction phase

Design model
Software components
Integrated software
increment
Test plan and procedure
Test cases
Support documentation
user manuals
installation manuals
description of current
increment

Transition phase

Delivered software increment
Beta test reports
General user feedback

Agile

- https://www.youtube.com/watch?v=yagk8a5w_4U

- Scrum

https://www.youtube.com/watch?v=WxiuE-1ujCM

Video Links

- Waterfall Model

https://www.youtube.com/watch?v=Y_A0E1ToC_I

- Prototyping Model

https://www.youtube.com/watch?v=dlxxXF0wrdc
https://www.youtube.com/watch?v=v8WzC5bCqY0

- Spiral Model

https://www.youtube.com/watch?v=6gMgXP1Sa9I