Process Model

Slide Set to accompany

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

by Roger S. Pressman

Quick Recap

- Software & characteristics
- Software Engineering – What and Why?
- Software Engineering – a layered Technology
- Why a layered Technology?
- A Generic Software Process
- Software Process Model vs Process Framework

Today Agenda

- Process Model
- Different Process Models
  - Waterfall Process Model
  - Incremental Process Model
  - Prototyping Process Model

Software Process Model

To solve actual problems in an industry setting, a software engineer or a team of engineers must incorporate a **development strategy**

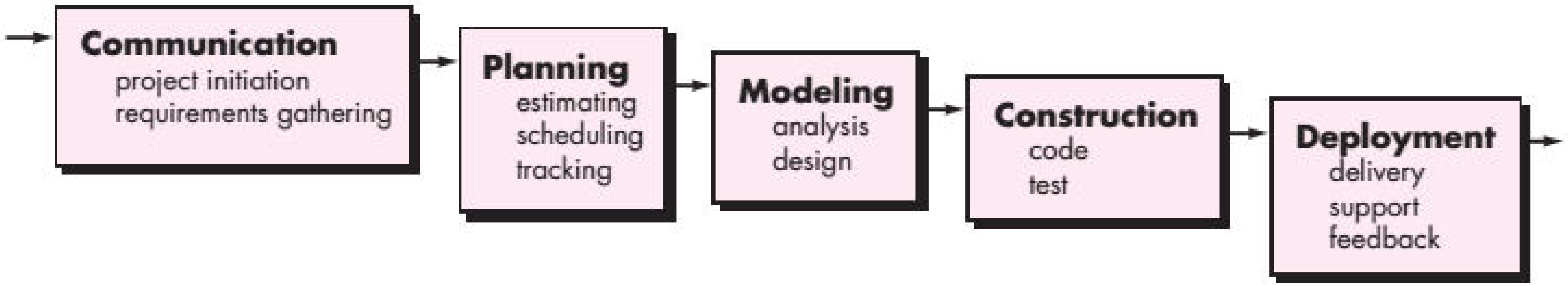
- This strategy is often referred to as **a *process model***

- A process model for software engineering is chosen based on:
  - -The nature of the project and application
  - -The methods and tools to be used
  - -The controls and deliverables that are required

<<<Numerous Process Models>>>

The Waterfall Model

- The linear sequential model is the oldest and the most widely used paradigm for software engineering



The Waterfall Model

- When Requirements are well understood
- When customer is capable to clearly communicate the requirements
- When well-defined adaptations or enhancements to an existing system must be made
- Real projects rarely follow the sequential flow that the model proposes
- Changes can cause confusion as the project team proceeds
- It is often difficult for the customer to state all requirements explicitly

The Waterfall Model

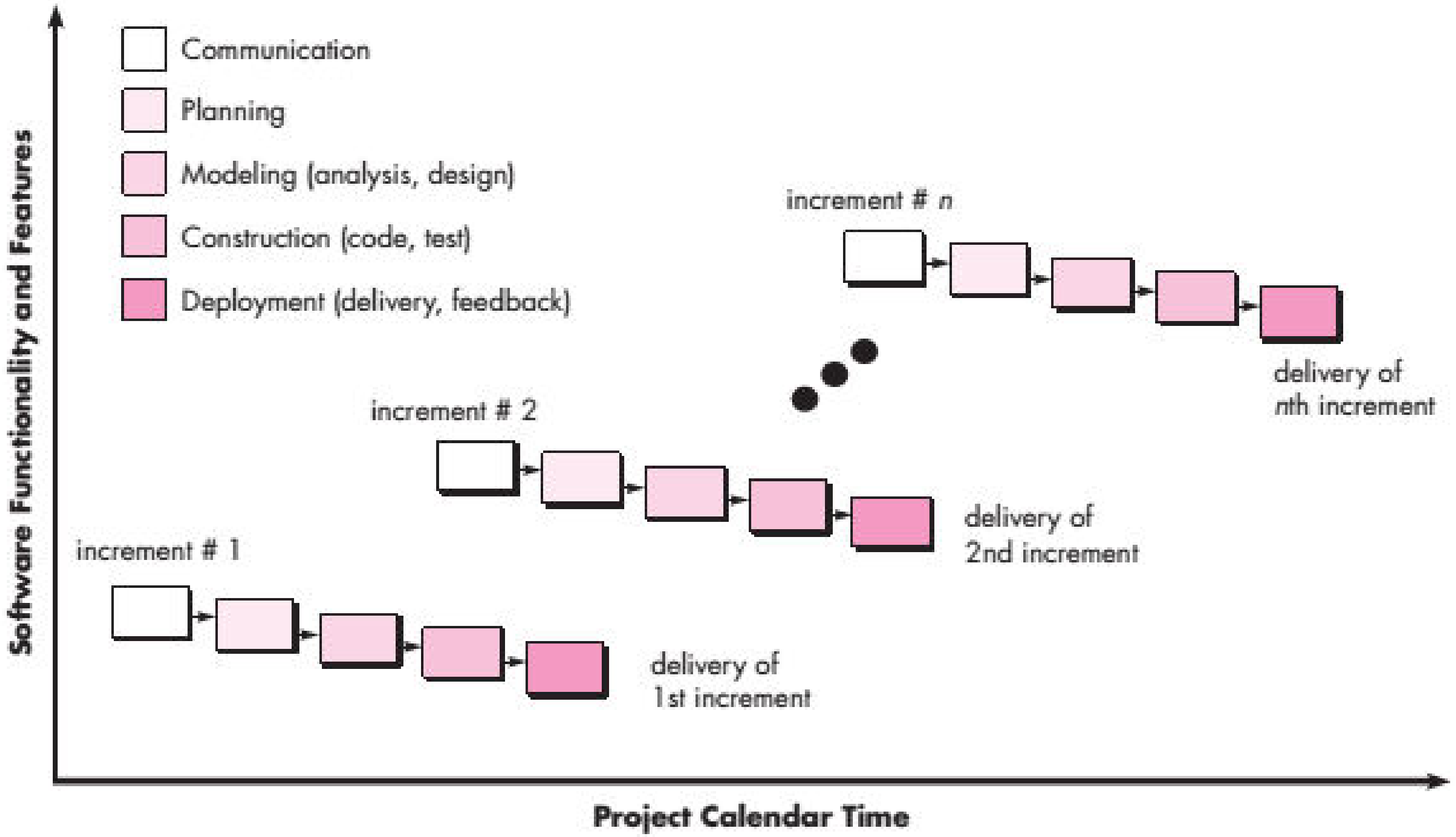
- The **customer must have patience**. A working version of the program(s) will not be available until late in the project time-span

- A **major blunder, if undetected** until the working program is reviewed, can be disastrous

- the linear nature of the classic life cycle leads to "blocking states" in which **some project team members must wait for other** members of the team to complete dependent tasks.

- In fact, the time spent **waiting can exceed the time spent on productive** work!

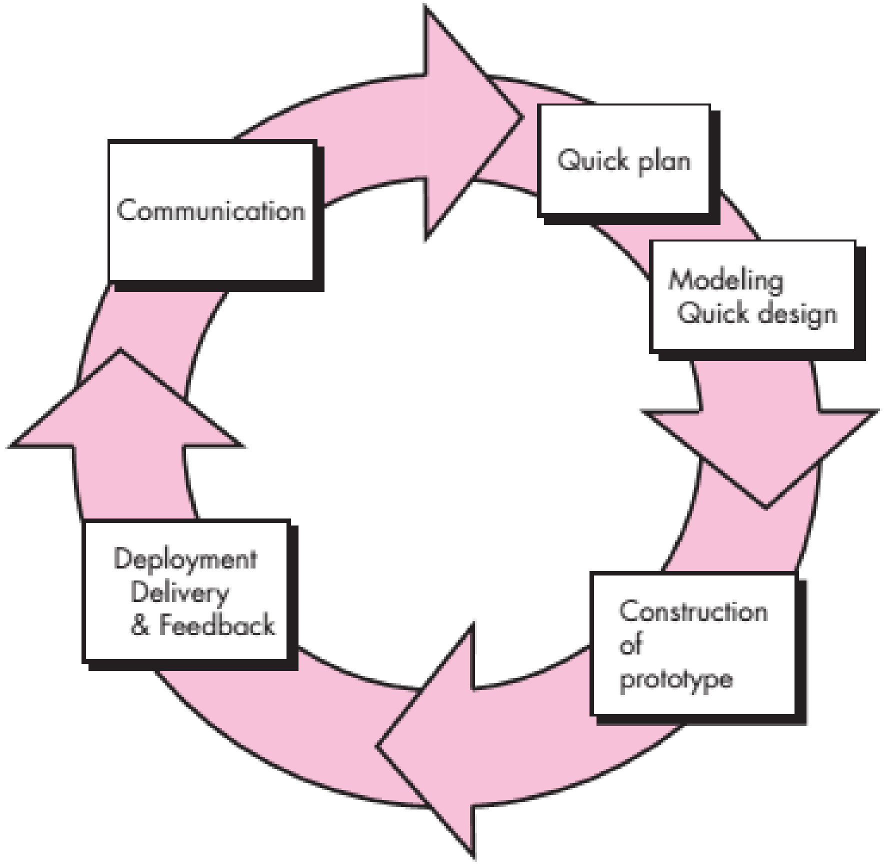
The Incremental Model



The Incremental Model

- Focuses on an **operational product** with each increment
- Provides a **needed set of functionality sooner** while delivering optional components later
- Used when requirements are well understood
- When Multiple independent deliveries are identified
- Useful also **when staffing is too short** for a full-scale development
- Work flow is in a linear (i.e., sequential) fashion within an increment

Evolutionary: Prototyping Model



Evolutionary: Prototyping Model

- Follows an iterative approach
- Used **when requirements are not well understood**
- Serves as a **mechanism for identifying software requirements**
- Focuses on **those aspects of the software that are visible** to the customer/user
- Feedback is used to refine the prototype

Evolutionary: Prototyping Model

- The customer sees a "working version" of the software, **wants to stop all development** and then buy the prototype after a "few fixes" are made

- Developers often make **implementation compromises** to get the software running quickly (e.g., language choice, user interface, operating system choice, inefficient algorithms)

- Solution:
  - **Define** the rules up front on **the final nature of the prototype** before it is built
  - In most circumstances, **plan to discard the prototype** and engineer the actual production software with a goal toward quality

Summary

- Process Model?
- Different Process Models
  - Waterfall Process Model
  - Incremental Process Model
  - Prototyping Process Model
- Selection Criteria for different Process Models?