Types of Software

- **Program**
  - Single executable
  - Used by very few people (maybe just the developer)

- **Product**
  - Used by many people
  - Polished
  - Used outside development environment

- **System**
  - Group of programs that work together
  - Complex interfaces/integration between parts

- **System Product**
  - Polish of a product
  - Multiple parts of a system
Need for Software Engineering

- **Problem with Scale**
  - Larger software → higher cost of coding, debugging, etc.
  - Larger teams:
    - Division of labor
    - More communication lines
    - Ensuring everyone on same page

- **Problem with Novelty**
  - Almost always new problem
  - Often “wicked problem”
    - Don’t know all issues until start working on it
Software Engineering Definition

- **Software Engineering**
  - **Design**
    - System/module/class/program/data design
    - Requirements
    - Architecture
  - **Development**
    - Coding
    - Testing
    - Debugging
  - **Management**
    - Work breakdown / team organization
    - Cost and resource estimation
    - Scheduling and workflow
    - Quality control
Software Development Metaphors

- **Penmanship** (“writing code”)
  - Too rigid
  - Not like real SE → suggests one person, little change, and originality
    - Real SE: multiple people, heavy change, reuse
  - Implies “throw one away” → wasteful

- **Farming**
  - Implies incremental approach → good
  - Implies lack of direct control
  - Details don’t quite make sense

- **Oysters**
  - Implies incremental approach
  - Leaves many details unspecified

- **Building**
  - Implies planning, preparation, and execution
  - Works with incremental approach
  - Allows variable scale and quality control
  - Includes concepts of:
    - Buy vs. build
    - Under- and over- planning
Software Construction focuses on the following:

- Detailed design
- Coding
- Debugging
- Integration
- Developer testing (unit testing and integration testing)
Why Software Construction?

- Large part of software development
- Central activity
- Can improve individual programmer’s productivity
- Source code → most accurate description of software
- Only activity guaranteed to be done