SE 2A04 Fall 2000

Software Development

Instructor: W. M. Farmer

Revised: 11 November 2001

Software Development Process

- A **rational** development process is needed to produce quality software
- Any proposed rational process is necessarily an **idealization**
  - Humans inevitably make errors
  - Communication between humans is imperfect
  - Many things are not understood at the start
  - Supporting technology always has limitations

Software Presentation

- Every software product should include documentation that presents the product to clients, reviewers, users, and maintainers
- It is useful to produce documentation that makes it appear as if the software product were developed by a rational process
  - Mathematicians have long followed this approach in presenting their results

See D. Parnas, "A rational design process: how and why to fake it", *IEEE Transactions on Software Engineering*, Vol. SE-12, No. 2, February 1986.

Development Activities

1. Investigation and specification of the **requirements** for the desired product
2. **Design** of a product that satisfies the requirements
3. **Implementation** of a product according to the design
4. **Verification and analysis** of the requirements, design, and implementation
5. **Maintenance** of the product (including the requirements and design documents)

1. Requirements

- What is the problem that needs to be solved?
  What are the product requirements that need to be satisfied?
- Output: **Requirements Specification**
  - Functional requirements
  - Requirements imposed by the environment
  - Other requirements (e.g., cost, delivery date, style considerations, performance)
- The Requirements Specification should include everything needed to design the product—no more, no less
  - Any design that satisfies the requirements should be acceptable

2. Design

- How will the problem be solved?
  How will the product requirements be satisfied?
- Output: **Design Document**
  - Includes a **module guide** that describes how the design is decomposed into modules
  - Includes a **module interface specification (MIS)** and **module internal design (MID)** for each module of the design
- The Design Document should include everything needed to implement the product—no more, no less
  - Any implementation that satisfies the design should be acceptable

3. Implementation

- What is a solution to the problem?
  What is an executable implementation of the design?
- Outputs:
  - **Source Code (with comments)**
  - **Product Description**
  - **Installation Instructions**
  - **User Manual**

4. Verification and Analysis

- What behavior does the product exhibit?
  Is the behavior correct?
- Forms of verification and analysis (outputs):
  - Inspection (**Inspection Reports**)
  - Testing (**Test Data**)
  - Mathematical verification (**Formal Mathematics**)

5. Maintenance

- What needs to be maintained?
  How will it be maintained?
- Outputs:
  - **Documentation and Software Repository**
  - **Maintenance Plan**
  - **Maintenance Records**
- The Repository should be maintained using a **version control system** such as CVS and should contain all previous versions of the documentation and software

Software "Life Cycle" Models

- Waterfall
- Refinement
- Incremental
- Spiral
- Prototyping

Documentation

- Good documentation is an essential part of good design
  - Documentation is to software engineering what medical records are to medicine
  - Documentation is usually not taken seriously by software developers, reviewers, and maintainers
- Like other kinds of engineering documentation, software documentation must be based on mathematics and logic
  - Concepts must be as clear and simple as possible
  - Notation must be both precise and concise

Documentation Recommendations

1. Design the documentation with great care
2. Use the same set of documents for the entire process
3. Integrate the documents with each other
4. Make the documents:
   (a) Accurate
   (b) Consistent
   (c) Easy to navigate
   (d) Easy to review
   (e) Easy to modify
5. Keep the documentation up-to-date and keep a record of all changes