CSE 403, Software Engineering
Lecture 2

Software Life Cycle

Announcements

- Quiz section will be held in CSE 305

Project Schedule

- Preliminary Design, April 15
- Preliminary Release, May 6
- Test Plan, May 20
- Design Critique, June 1
- Final Release, June 1

Writing assignment

- Due Monday, 1:30 pm, April 4
- Individual Assignment
- Target length: two pages

Critique the Surgical Team model proposed by Brook's as an organization for your GizmoBall project. You should first describe how you would adapt the model to a 6 or 7 person team, and then evaluate how appropriate it would be as a team organization.

Lecture schedule

- 1. Introduction
- 2. Life Cycle
- 3. Teams
- 4. Risk analysis
- 5. Requirements and Design
- 6. Development and Deployment

Course goal

- To gain an understanding that developing a software product is not merely a matter of programming
If it's not merely programming

- What is it?

**Life Cycle**

Software life cycle

**Life Cycle (McConnell)**

- System specification
  - Requirements Analysis
  - Architectural Design
  - Detailed Design
  - Coding and Debugging
  - Unit testing
  - System testing
  - Maintenance

**Model Goals**

- Understand what goes on
- Organize workflow
- Formal process
Waterfall Model

- Strong directionality in stages
- Limited up stream interaction
- Very large costs in fixing errors arising from early stages

Critiques of the Waterfall Model

Spiral Model

- Understand process
- Defining procedures
- Decomposing workflow
- Track, clarify, modify requirements through life cycle
- Management tool

Limitation of models

- A model is just a model
- Artificial constraints
- Compromises with model necessary
  - (as with almost everything else in SE)
- Risk of overemphasizing process
  - The process is not the end in itself
  - Product delivery is

Requirements on requirements

- Who are they for?
- What are they for?
  - Pitch to management
  - Fodder for market study
  - Basis for legal contract
- Easy to understand, concise, complete, unambiguous, . . .
Requirements

- “Gather and document the functions that the application should perform for the users in the users' language and from the users' perspective"
- Requirements should neither constrain nor define methods of implementation

Customers

- (Almost) every large software project has a customer who is paying the bills
- Project requirements driven by this customer