Lecture 11: Working in Groups

CS190: Software System Design

March 6, 2002
Steven P. Reiss

I. Today's Class

A. As I said initially CS190 is not a programming course per se

1. More a course on how to organize a large project
2. And that organization is as much people as code

B. A lot of work has been done in SE regarding working in teams

1. I want to cover some of that
2. And then you can have group meetings

II. Mythical Man Month

A. Working in groups is difficult

1. Productivity decreases
   a) 5000 lines/person/year alone
   b) 4000 lines/person/year in teams of 5
   c) 3000 lines/person/year in teams of 9
2. Time is spent on interaction
   a) Up to 50% of ones time can be spent this way
3. Time is spent on misunderstandings
4. Personality clashes can abound

B. How to measure the cost of a project

III. Group Organization

A. Key to productivity is group organization

1. To minimize interaction
2. To maximize productive individual time
3. To facilitate cooperation
4. To minimize misunderstandings

5. **To minimize conflict**

B. Different organizations are possible

1. **Unorganized -- n(n-1) communications channels**
2. **Chief Programmer team**
   a) Chief programmer (leader)
   b) Librarian (documentation manager)
   c) Programmers
   d) O(n) communications channels

C. Chief Programmer Team

1. **Team Leader**
   a) Experienced and highly qualified
   b) Able to understand the whole system
   c) Able to take responsibility
   d) Able to judge cost/time of components
2. **Librarian**
   a) Takes responsibility for communications
   b) Facilitiates interaction through documentation
   c) Ensures team members communicate appropriately
3. **Programmers**
   a) Responsible for compartmented pieces of the system
   b) Interact via documentation/standards

D. Organizing an XP Team

1. **Not much different from standard team**
2. **Except**
   a) Programmers have responsibility for features not modules
   b) Documentation is typically more dynamic
   c) Test cases need to be managed as well
3. **Leader is responsible for**
   a) Assigning programmers to features
   b) Steering development
   c) Choosing next set of features

d) Deciding when to do refactoring

IV. Project Organization

A. As important as group organization

1. Project should be broken down to minimize communications

2. Individuals should have minimum dependencies on others

a) Facilitates writing code

b) Facilitates testing and debugging

c) Enhances productivity

B. This is an aspect of design

V. Designing for Groups

A. Minimize dependencies

1. Core + Extensions design does this nicely

2. Compartmentalize (facade design pattern)

3. Centralize functionality

4. Heavy use of interfaces

B. Determine dependencies early

C. Define the dependent interfaces first

1. By analyzing dependencies

2. Make the dependencies explicit

D. Develop high-level coding conventions

1. To avoid name conflicts

2. To make coding and naming easier

VI. XP for Groups

A. Design interfaces so that functionality is orthogonal

1. New features should be independent of old features where possible

2. New features should be independent of each other

B. Choose new features to minimize overlap

C. Coordinate the need for refactoring

1. Someone should take charge to decide when things need to be changed