SE Principles

Topics for Today

• Review of the “initial”
• SE Principles
• Readings for next week
Typical Software Development

- Concept Formation
- Requirements Specification
- Program Design
- Program Implementation

The Need for Disciplined Practices

- The job of software engineers is to
  - Produce quality products
  - Produce them on schedule
  - And do this work for the planned costs

- In this class I hope we will learn some of this
  - You will also need a lot of practice!!
Overview

• Software engineering is based on a collection of fundamental principles
• These principles guide the development of all aspects of software development
  ♦ Languages
  ♦ Methods
  ♦ Tools
  ♦ Process
  ♦ Project management

How it All Relates
We Will Cover

- Rigor and formality
- Separation of concerns
- Modularity
- Abstraction
- Anticipation of change
- Generality
- Incrementality

Rigor and Formality

- Software development is a creative process
  - Creativity implies informality and chaos
- Rigor and formality seems to contradict creativity
  - Not necessarily so
    - Increase the confidence in the creative results
- Evident in
  - Programming languages, design notations, requirements specifications, process definitions
Separation of Concerns

- We cannot deal with all aspects of a problem simultaneously
  - One way to conquer complexity
- Separate issues and tasks
  - Separate functionality from efficiency
  - Requirements specification from design
- Various types of separation
  - In terms of time
  - In terms of qualities
  - In terms of views of an artifact

Modularity

- A complex system must be broken down into smaller modules
- Three goals with modularity
  - Decomposability
    - Break the system down into understandable modules
    - Divide and conquer
  - Composability
    - Construct a system from smaller pieces
    - Reuse, ease of maintenance, OO frameworks
  - Ease of understanding
    - The system will be changed; we must understand it
    - Understand in pieces versus understanding the whole
More Modularity

Two Essential Properties

- High Cohesion
- Low Coupling
Abstraction

• Identify the important aspects and ignore the details
• Must have different abstractions of the same reality
  - Provide different views
• Examples of abstraction
  - Design notations
  - Project planning
  - Etc.
• Two key concepts
  - Information hiding and data encapsulation

Anticipation of Change

• Change is inevitable
  - We might as well plan on it!
• This effects all aspects of Software Engineering
  - Make sure all artifacts are easy to change
    - Modularization and separation of concerns
  - Make sure you can maintain many versions of all artifacts
    - Configuration control
  - Plan for personnel turnover
  - Plan for a rapidly changing market
  - Plan for rapidly changing technology
Generality

• In every problem, attempt to find a more general solution
  ◆ General problem is often easier to solve
  ◆ A generalized solution may be reusable
  ◆ If you are lucky, you may even be able to buy instead of build

• Attempt of the software industry to parallel hardware and manufacturing

Incrementality

• Move towards the goal in increments
  ◆ Very difficult to have a “big bang” approach to anything

• Areas where we see incrementality
  ◆ Identify useful subsets of an application and deliver in increments
  ◆ Prototyping
  ◆ All software development adds functions a few at the time
  ◆ The whole development process; the spiral model
Concluding Remarks

- These fundamental principles guide all aspects of software development
- Keep them in mind during the class and when
  - Planning a new software project or product
  - Design your software
  - Evaluate new tools and technologies
  - Evaluate new methodologies and management techniques
- Remember
  - Tools, methodologies, and techniques will evolve, the principles remain the same

We Have Learned

- We need to revisit a few topics related to requirements
- You need to go over basic OO modeling
- Some fundamental principles
- Next time
  - Requirements: What, why, and how?
- Readings
  - Rich traceability
  - The WRSMP model
Original Engine System

Solution - Automate the Process