Software Design, Practice, and Experience (SDPE) Task Force

Summary
October 2004
Dr. Paul Grabow, chair

Members
- Bill Booth
- Jeff Donahoo
- Roxana Girju (research leave)
- David Sturgill
- Paul Grabow (chair)
Charge

- Consider software design, engineering, development, implementation, etc.
  - Determine a list of critical and important concepts/skills related to these areas
  - Determine the depth to which the critical and important concepts are covered in the rest of the curriculum

Charge, cont.

- Create a two-course sequence which, in concert with other classes in the curriculum, covers the critical and important concepts as completely as possible
- Provide a list of “left out” concepts
- Provide a timeline for accomplishing these objectives to the curriculum committee by March
2004 Timeline

- Apr 15th
  - charge, strategy
- Apr 20th
  - state of software practice, IEEE/ACM curriculum documents
- Apr 28th
  - current curriculum, list of possible concepts
- May 13th
  - main themes/categories
- Sep 15th
  - Software I objectives, Version 1.0
- Oct 6th
  - Software I objectives, Version 2.0
- October 13th
  - Software II objectives, Version 1.0

Related Existing Courses

- CSI 3342 Introduction to Software Design
  - Object-oriented design methods
  - Iterative development, e.g., Unified Process
  - UML diagrams
  - Small group project, e.g., 3 students per group

- CSI 4344 Object-oriented Development
  - Design patterns
  - Message-based design
  - Multi-platform implementation
  - Class project, e.g., 9 students arranged into three groups
Possible New Courses

- Software I
  - Sophomore-level

- Software II
  - Junior-level

Software I objectives, version 2.0

- **Language:** know how to use a 2nd programming language, e.g., Java, and its
  - IDE
  - Debugger
  - API

- **Design Notation:** know how to interpret and use
  - UML class and sequence diagrams
  - Use cases

- **Methods/Strategies:** know and use methods/strategies for
  - Debugging
  - Testing
  - Error handling
Software I objectives, cont.

- **Standards:** know and apply standards for
  - Coding

- **Development Process**
  - Understand and apply simple lifecycle model
  - Interpret deliverables, e.g., requirements, system specification
  - Create and execute Test plan

- **Design Concepts:** understand and apply
  - Design patterns
  - Coupling and cohesion