#### **General Introduction**

#### CPRE 416-Software Evolution and Maintenance-Lecture 1

### **General Information**

Office: 3132 Coover Hall

Office Hours: Monday 2:30-3:30 and Friday 1:10-2:10

Email: kothari@iastate.edu

Phone: 294-7212

**Appointments:** Make talk appointments by talking to me before or after the class. Appointments are needed during the scheduled office hours.

**Emails:** Minimize emails. Avoid sending emails to make appointments or to conduct discussions.

### Concern

Although software maintenance and evolution account for the majority of the costs involved in a software system's life cycle, the amount of time spent on these topics in a typical computer science or software engineering curriculum appears to be negligible.

## **Course Objectives**

- Learn about software evolution and maintenance by performing following activities:
  - Explain important aspects of software evolution and maintenance.
  - Identify and apply appropriate program reading strategies.
  - Use different types of program comprehension tools.
  - Evaluate program analysis techniques and tools.
  - Perform defect analysis and quality inspections of software.
  - Gather, analyze, and present information on technical topics.

## Topics

- Aspects of software evolution and maintenance such as program comprehensions, defect analysis etc.
- Process and planning such the IEEE/EIA 12207 life cycle process.
- Tools such as program comprehension and defect analysis tools.
- Current research such as concern-based program analysis and software visualization.

### Aspects of Software Evolution and Maintenance

- Program comprehension
- Code inspection
- Defect analysis
- Refactoring
- Code Migration
- Impact analysis

## Learning Activities

- Discussion of foundational material.
- Gathering information from the Internet.
- Goal-directed reading of expository and research papers.
- Empirical studies with source code and tools.
- Critical analysis of the proposed problem solving techniques.
- Presentations on project work.

## Reading Research Papers

- Reading research papers effectively is challenging.
- What are the motivations for this work?
- What is the proposed solution?
- What is the evaluation of the proposed solution?
- What are the contributions?
- What are future directions for this research?
- Often desirable to read the paper ``out of order''.

### Wicked Problem

- Software development is often described as a "wicked problem" because of:
  - lack of precise problem definition
  - the difficulty of verifying and validating proposed solutions
  - the inadequacy of methods and methodologies in guaranteeing the quality of the result
  - essential uniqueness of each system
  - solutions are good/bad, not true/false

## **Problem Solving**

- Complexity is added rather than reduced with increased understanding of the problem.
- Good abstractions are must a must effective problem-solving
- A breadth of experience is necessary so that similarities and differences with past strategies can be used to deal with new situations.

## **Problem Based Learning**

- I will emphasize problem-based learning:
  - A problem serves as a guide for learning.
  - Attack the problem by articulating questions that break it into smaller manageable problems.
  - focus on the process of knowledge acquisition rather than the products of that process
  - Learning of content and skills in a collaborative environment.
  - Look for appropriate generalizations that are firmly anchored in real-world characteristics of the problem.
- I will serve more as facilitator and less as instructor.

# Using Electronic Library

- IEEE Xplore: <u>http://www.lib.iastate.edu/collections/db/ieeexx.html</u>
- Process:
  - Suppose you get following reference after searching on Google <u>http://portal.acm.org/citation.cfm?id=837837</u>
  - Google search shows that the paper appeared in International Workshop on Program Comprehension (IWPC) in 96.
  - Click on Xplore, click on conferences, then type IWPC in the search box and go.
  - You will get a yearly listing of all IWPC proceedings.
  - Click on the appropriate year, the Table of Content comes up.
  - Click on the PDF link for the paper.
- WCRE is another conference with several relevant papers for this course.

## Grading

- Will be based on:
  - projects 45%, homework 45%, and other (attendance, class participation, exemplary work etc.) 10%
- Grading will be pretty close to a straight scale with A: 91% and above, B: 81% -90%, etc.

## Projects

- Three or four projects. It will involve:
  - empirical studies
  - design and analysis of problem solving techniques
  - Gathering information from the internet and library
  - reading research papers
  - goal-directed reading of large software,
  - defect analysis of software,
  - reports, papers, presentations, and demos.
- Group activity. Submissions:
  - presentation and/or demo one per team (email)
  - log of work and comments on learning experience per individual (hardcopy)

## Weekly Homework

- It will involve:
  - Gathering information from the internet and library
  - reading and analysis of research papers
  - Small code reading exercises
- Individual activity. Submissions:
  - hardcopy at the beginning of the class on due date
  - Must be a well-written Word document
  - Follow instructions (e.g. use a standard template that includes your name, assignment dates etc.)

### **General Instructions**

- Attendance is mandatory.
- Inform if you are going to miss a class.
- Participate in class discussions.
- Check course website regularly.
- Follow submission guidelines.
- Keep regular contact with the instructor for help, questions, concerns, or comments.

## **Exemplary Work**

- These are some examples:
  - Participation: Find and send interesting information with your comments to explain its relevance, critical analysis of information presented in the class.
  - Homework: extremely well-written document, innovative answers, references and comments on other highly relevant material.
  - *Projects*: creativity and originality (e.g. an empirical study that exposes interesting ideas), very interesting presentation or a demo, novel research ideas.

### About Myself

- Background: Ph.D. in mathematics from Purdue, have taught in mathematics, electrical engineering, and computer science departments, worked at AT&T Bell Lab.
- **Professional Activities:** teach courses in software engineering, operating systems, and parallel computing, research with three Ph.D. students on software tools, write and review research papers and proposals, work with companies on applications of research, chair the software systems group, develop undergraduate degree program in software engineering, and run a company at ISU research park.

### What I would Like

- I would like to learn about your interests and concerns and get constructive feedback so that I can direct my teaching and work with you to enhance learning.
- I would like you to focus on developing problem solving and communication skills and enjoy learning.
- I would like you to be engaged in discussions and active learning.
- I would like you to spend significant time on course work on a regular basis.

### About Yourself

- Submit one page.
- I would like to know about:
  - Courses you have had and the ones you liked and why.
  - Experience with Co-Ops
  - Experiences if any of working with large software – understanding, debugging etc.
  - Any comments and questions you would like to share.

### Get Started

- Read the course website thoroughly.
- Explore the ISU electronic library.
- Form a team for the first project and send me the names, email addresses, and primary contact person.
- Play with the Wink software. You will need it to create the demo for your first project.