



CS 497
Software Engineering I

Fall Semester 2006
Chris Lüer, PhD

1. What is this Class about?

- You know:
 - How to code
 - on your own
 - small programs
 - At least one programming language
 - Algorithms and data structures
- You will learn in this class:
 - How to develop software
 - that is large and complex
 - in a team
 - Tools and methods used in industry
 - “Software Engineering”
 - UML, Design Patterns, processes

© 2006. 497-1-2

Software Engineering

- Practical problems of software development
 - especially in large, complex projects
- Related to management and psychology
- Focus on activities other than implementation
- Subareas:
 - Requirements Engineering
 - Software Architecture, Design
 - Quality Assurance, Testing
 - Configuration Management
 - Formal Methods
 - Software Engineering Environments
 - Human-Computer Interaction
 - Project Management

© 2006. 497-1-3

Good Software

- Software Engineering is about making good software
- Maintainability
 - changing user needs
 - new technologies
- Dependability
 - software in hospitals, airplanes
- Efficiency
 - distributed systems
- Usability
 - graphical user interfaces
 - easy to learn for new users

© 2006. 497-1-4

Challenges of Software

Engineering

- Legacy software
 - old software
 - maintaining it, interfacing with it
- Heterogeneity
 - Internet
 - wireless networks
 - small devices: PDAs, embedded computers
- Timeliness and Cost
 - getting what you need right away
 - and cheap, too!

© 2006. 497-1-5

What is this Class about? –

Methods

- Processes
 - iterative processes
 - extreme programming
- Requirements Capture
- Analysis
- Design
- Testing

© 2006. 497-1-6

What is this class about? —

Notations

- UML
 - Class diagrams
 - Sequence diagrams
 - Collaboration diagrams
 - Use case diagrams
 - State diagrams
 - Activity diagrams

© 2006. 497-1-7

What is this class about? —

Tools

- JUnit / CPPUnit
 - unit testing
- Rational Rose
 - UML
 - Commercial
 - Reverse engineering
- Argo/UML
 - UML
 - Open source
- CVS
- Development Environments
- 498:
 - testing tools
 - scripting languages

© 2006. 497-1-8

What is this class about? —

Project

- Customer
 - gives requirements
 - signs off on progress at the end of 497
 - signs off on finished product at the end of 498
- Team work
 - coordination
 - communication
- Management
 - time planning
 - decision making

© 2006. 497-1-9

Prerequisites

- CS 336/436 Databases
- CS 232 Data Structures

© 2006. 497-1-10

Related Classes

- Good if you took them, good if you are taking them now
- CS 337 Internet Programming
- CS 345 Graphical User Interfaces
- CS 397 Web Technology
- CS 438 Computer Graphics

© 2006. 497-1-11

1.2 Administrative Stuff

- Instructor
 - Chris Lürer (clueer@bsu.edu)
 - Office hour: Tu 2:00-3:30, Th 5:00-7:00
- Teaching assistant
 - Tai-Lin Han (tahan@bsu.edu)
- Check the Web page frequently!

© 2006. 497-1-12

Textbooks

- Pressman
 - broad
 - background
 - all of software engineering
- Fowler
 - UML
 - Technical, in-depth view
- Handouts
 - design patterns
 - as needed

© 2006. 497-1-13

Be Involved

- Class Web site
- Student Web sites
 - Team Web sites
- Mailing list: CS497-L@listserv.bsu.edu
- Be active
 - Ask questions
 - Participate in class discussions
 - Cooperate with team members
 - Go to office hours

© 2006. 497-1-14

Cheating

- Letter to Dean of Students
- Course grade lowered, possibly to F
- No team work on homeworks
- Things that are copied from books or Web pages need to be quoted and the source must be given

© 2006. 497-1-15

Grading

- Final Exam 25%
- Midterm Exam 10%
- Project 40%
- Homeworks 20%
- Quizzes 5%
- Peer evaluations for project

© 2006. 497-1-16

Who Are You?



© 2006. 497-1-17

Simple Schedule

The screenshot shows a spreadsheet window titled "Schedule.xls" with a table containing the following data:

	1	2	3	4	5	6	7
1	Feature	Task	Priority	Orig Est	Curr Est	Elapsed	Remain
2	Spell Checker	Add Menu Item	1	12	8	8	0
3	Spell Checker	Main Dialog	1	8	12	8	4
4	Spell Checker	Dictionary	2	4	4	4	0
5	Grammar Checker	Add Menu Item	1	16	16	0	16
6							
7							
8							

- use spreadsheet
- hours
- additional column: person assigned to
- feature: something from requirements document, e.g. use case

© 2006. 497-1-18

Which programming language to
choose?



© 2006. 497-1-19



© 2006. 497-1-20

1.3 Architecture of a Project

- Architectural decisions
 - must be made at the beginning of the project
 - cannot easily be reversed later
 - are important for success
- Examples
 - which programming language?
 - which platform?
 - MS Windows, Unix, Web?
 - does it have a Gui?
 - does it have a database?
 - what libraries are used?
 - is a three-tier architecture used?

© 2006. 497-1-21

Architectural Constraints in this Class

1. Programming language
 - use an object-oriented system programming language unless there is a very good reason not to
 - C++, Java, C-Sharp
2. Architecture
 - a three-tier architecture is required if applicable
 - presentation layer
 - business logic layer
 - persistence layer
 - all three layers are usually needed

© 2006. 497-1-22