



Course Overview

CS 577a

Software Engineering I

Barry Boehm, USC

Dan Port, USC

David Klappholz, Stevens

Fall 2002



Outline

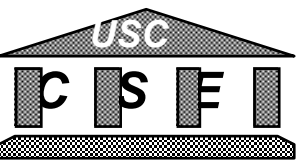
- **Software Engineering Definition and Learning Objectives**
 - Comparison of CS 510 and CS577a
- **Class Overview**
 - Class Project Objectives and Approach
 - Top 10 Risk Items



“ Software Engineering:” The disciplines which distinguish the coding of a computer program from the development of a software product.

Issues \ Stages	Requirements, Architecture	Design, Code	Test, Implement, Maintain
Computer Science		CS Focus	
User Applications			
Economics			
People			

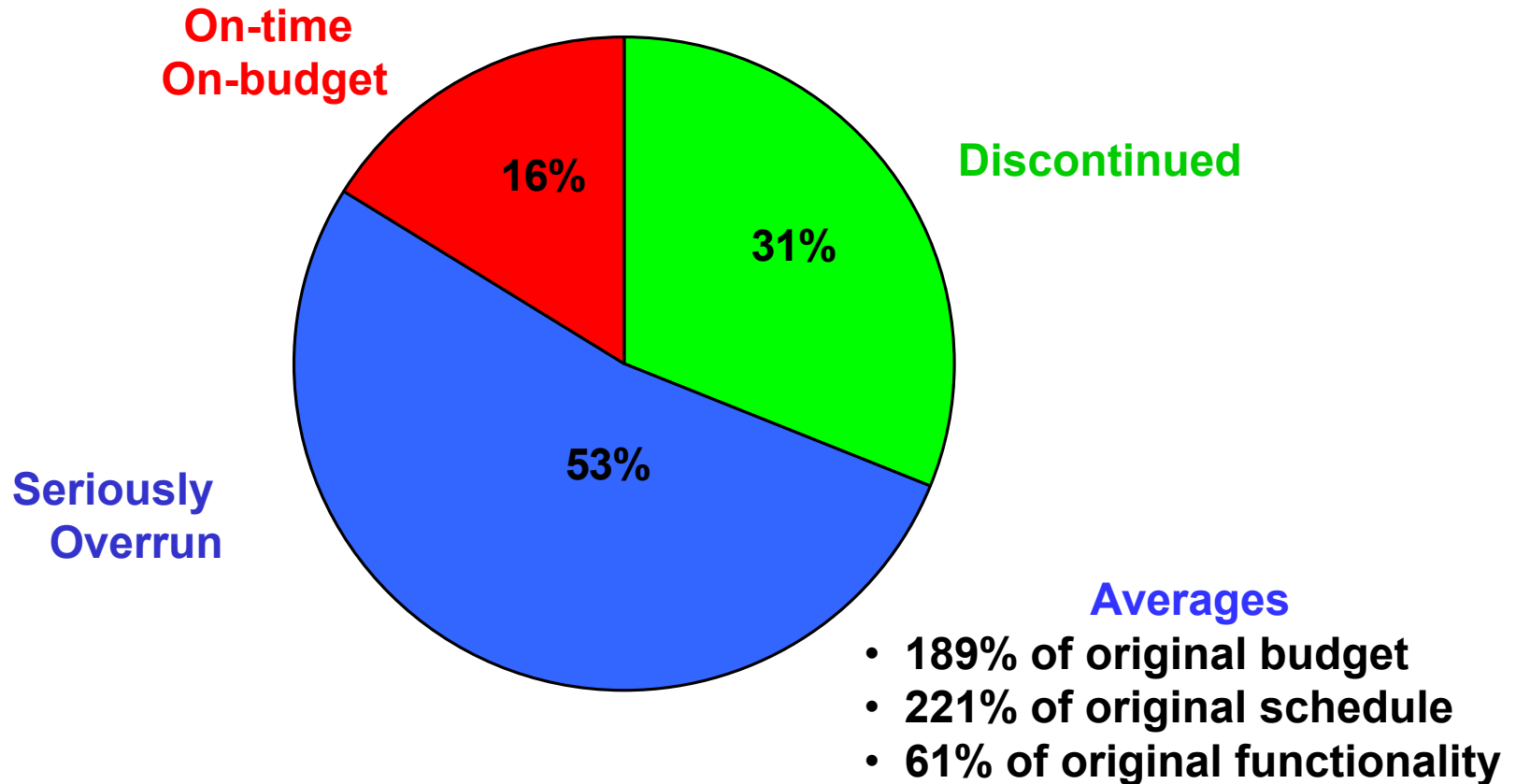
- **Accommodate new tools and techniques**
 - **Web browsers, GUI prototypers, WinWin, Spiral processes**
- **Integrate all these considerations**
 - **Via integrated models (MBASE)**

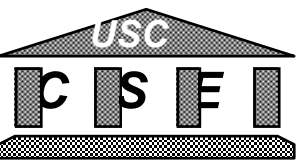


A Short History of Software Processes - necessarily oversimplified

Decade	Orientation	Example(s)	Problems
1950's	Engineering	SAGE	Engineering orientation
1960's	Programming	Code and fix	Rework, scalability
1970's	Requirements	Waterfall	Risk, GUI, COTS/reuse
1980's	Many	Evolutionary, Incremental, Spiral, Helix, JAD, RAD	Overgeneralization/ Overspecialization
1990's	Emergence	Win-Win Spiral, Rational, Adaptive (model generators)	Value and economics
2000's	Value	Benefits realization, MBASE	Integration with emergence

Software Engineering Is Not Well-Practiced Today -Standish Group CHAOS Report 1995





Information Technology Trends

Traditional Development

- Standalone systems
- Stable requirements
- Rqts. determine capabilities
- Control over evolution
- Enough time to keep stable
- Repeatability-oriented process, maturity models

Current/Future Trends

- Everything connected (maybe)
- Rapid requirements change
- COTS capabilities determine rqts.
- No control over COTS evolution
- Ever-decreasing cycle times
- Adaptive process models



Traditional vs. Emerging SW Processes

Traditional

- **Contract-oriented**
 - Your problem; win-lose
- **Sequential**
- **Procedure-driven**
- **Programming-driven**
- **Many interlocking milestones**
- **Focus on discipline & control**

Emerging

- **Collaboration-oriented**
 - Our problem; win-win
- **Cyclic, concurrent**
- **Risk-driven**
- **Reuse-driven**
- **Critical anchor points with subsidiary milestones**
- **Mix of flexibility & discipline**



Comparison of CS 510 and CS 577a/591a

CS 510

- **COCOMO II Extensions**
- **Microeconomics**
 - Decision Theory
- **Rapid Application Development**
- **People Management**
- **Final Exam**

- **MBASE, VBSE**
- **WinWin Spiral**
 - Risk Management
- **Planning & Control**
 - COCOMO II
- **Business Case Analysis**

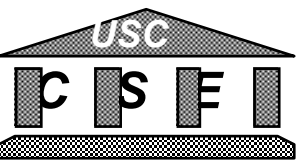
CS 577a/ 591a

- **S/W - System Architecting**
- **Operational Concept & Rqts. Definition**
 - WinWin System
 - Prototyping
- **OO Analysis & Design**
 - Rational Rose
- **Team Project (591a: IV&V)**



Class Project Objectives

- **Create life-cycle architecture and plan for developing an**
- **E-Services capability for the**
- **USC Information Services Division**
 - plus a few supplementary projects



Class Project Artifacts

- 1. An Operational Concept Definition**
- 2. A Prototype of Key System Features**
- 3 A System Requirements Definition**
- 4. A System and Software Architecture Definition**
- 5. A Life Cycle Plan**
- 6. A Feasibility Rationale, assuring the consistency and feasibility of items 1-5**



Class Project Approach

- **Artifacts developed concurrently**
 - Using Win-Win Spiral approach
- **Two critical project milestones (anchor points)**
 - Life Cycle Objectives (LCO)
 - Life Cycle Architecture (LCA)



Team Structure

- **Five-person teams**
 - Ops Concept and Requirements team members also develop Life Cycle Plan and Quality Plan
 - **Feasibility Rationale member acts as Project Manager**
 1. Ensures consistency among the team members' artifacts (and documents this in the Rationale).
 2. Leads the team's development of plans for achieving the project results, and ensures that project performance tracks the plans.
- Teams formed by Monday, Sept. 9
- Web questionnaires should help in team formation
- **Start forming teams now!**
 - What are your skills? What roles would you like to have?



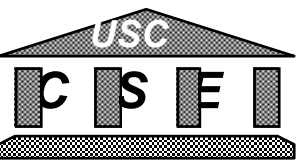
Individual Responsibilities

- **Homework assignments**
- **Quiz at end of each lecture**
 - 5 min. question on anything presented
 - 5 points, lowest 3 dropped, no “make-ups”
- **Contribute to team project**
 - 40 points assigned at end of course
- **Individual Critique**
 - 160 points, due after LCA packages submitted
- **Effort reports, Review other peoples products**
- **Presentation at two project reviews (ARB’s)**
- **Acquire Brooks “Mythical Man Month”**
 - USC bookstore, Amazon, Barnes and Noble, etc.
 - Must be completely read by 9/4 and is needed for HW#1



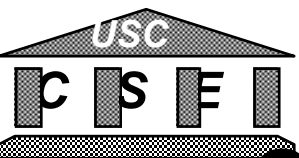
WinWin User Negotiations

- **Teams work with representatives of USC e-services users**
 - Art, cinema, engineering, business, etc.
 - E-services infrastructure
 - Begin with problem statements of top digital library needs
- **Use Easy WinWin system to balance user needs with customer and developer win conditions**



Major Class Project Milestones

September 9	--	All teams formed
September 20	--	Initial Prototypes/Storyboards
September 27	--	WinWin Negotiation Results
October 9	--	LCO Drafts on Web Site
October 15-21	--	LCO Architecture Reviews
October 25	--	LCO Package Due
November 13	--	LCA Drafts on Web Site
November 20-26	--	LCA Architecture Reviews
December 2	--	LCA Package Due
December 9	--	Individual Critiques Due



CS 577a, Fall 2002 Questionnaire

Please fill out and return.

Name: _____

Student ID #: _____

Dept./Degree Program: _____

Job, Employer: _____

Software Work Experience (years): _____

Phone, fax numbers: _____

E-mail Address: _____

Acknowledgement: I acknowledge the importance of USC's academic integrity standards (with respect to plagiarism, referencing others' work, etc.), and agree to abide by them.

Signature: _____



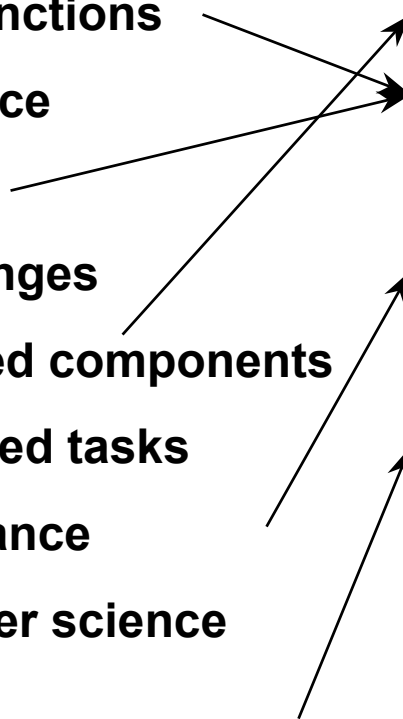
Top 10 Risk Items: 1989 and 1995

1989

1. Personnel shortfalls
2. Schedules and budgets
3. Wrong software functions
4. Wrong user interface
5. Gold plating
6. Requirements changes
7. Externally-furnished components
8. Externally-performed tasks
9. Real-time performance
10. Straining computer science

1995

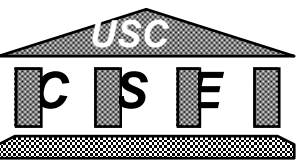
1. Personnel shortfalls
2. Schedules, budgets, process
3. COTS, external components
4. Requirements mismatch
5. User interface mismatch
6. Architecture, performance, quality
7. Requirements changes
8. Legacy software
9. Externally-performed tasks
10. Straining computer science





Primary CS577 Risk Items

- **Personnel: commitment; compatibility; ease of communication; skills (management, Web/Java, Perl, CGI, data compression, ...)**
- **Schedule: project scope; IOC content; critical-path items (COTS, platforms, reviews, ...)**
- **COTS: see next chart; multi-COTS**
- **Rqts, UI: mismatch to Library user needs**
- **Performance: #bits; #bits/sec; overhead sources**
- **External tasks: Client/Operator preparation, commitment for transition**



COTS and External Component Risks

- **COTS risks: immaturity; inexperience; COTS incompatibility with application, platform, other COTS; controllability**
- **Non-commercial off-the shelf components: reuse libraries, government, universities, etc.**
 - **Qualification testing; benchmarking; inspections; reference checking; compatibility analysis**