

CSC 4330: Software Systems Development

Credit Hours: 3 hours

Prerequisites:

CSC 3102: Advanced Data Structures

Prerequisites by Topics:

Programming fundamentals, data structures, functional decompositional

Catalog Course Description:

Software requirements analysis; design representation, programming methodologies; verification, validation, maintenance and software planning.

Course Outcomes

1. Master attributes and assessment of quality, reliability and security of software,
2. Master principles of software development process,
3. Master process selection regarding software development,
4. Master the understanding and implementation of a software development process,
5. Master the understanding of domain analysis,
6. Master the understanding of requirements specification documents,
7. Be exposed to technical writing and oral presentations.
8. Master the notion of user-centric viewpoints in system design,
9. Master the modeling of static attributes of a system design,
10. Master the modeling of the dynamic attributes of a system design,
11. Master the role of management in software development,
12. Master working on a team in a moderate-size software development project,
13. Master presentation of technical development concepts orally to a group,
14. Be familiar with formal specification notations,
15. Master the knowledge and comparison of various testing strategies.
16. Be familiar with the difficulties of working in teams and use of strategies to overcome those difficulties.

Texts and Other Course Materials

Pfleeger, S. L. Software Engineering: Theory and Practice, Third Edition, Prentice Hall.

Booch, Grady, et al. The Unified Modeling Language User Guide, Addison-Wesley.

References

Fowler, Martin. UML Distilled: A Brief Guide to the Standard Object Modeling Language -- Second Edition

Hutcheson, Marnie L. Software Testing Fundamentals : Methods and Metrics, Wiley.

Additional readings from other books, websites, technical reports, conference proceedings, and journal articles.

Major Topics

- Software process, methods and tools,
- Software process models, including the waterfall model, the incremental model, and the spiral model, prototyping models,
- Object-oriented methods using UML models,
- Management concepts such as team organization (chief programmer team, democratic team and manager team), planning and scheduling (milestones, PERT charts, Activity charts and staff allocation),
- Factors that affect the quality of software and software quality assurance,
- Process, project and software metrics,
- Software project planning including software scope, estimation of resources, project cost estimation, decomposition techniques and empirical estimation models,
- Requirements analysis, elicitation, and specification,
- Specification methods models including system models, functional specification models, non-functional specifications,
- Formal specifications: axiomatic, functional, and algebraic specifications,
- Traits of good technical writing,
- Data modeling, functional modeling, behavioral modeling and structured analysis,
- Design concepts including abstraction, refinement, modularity, software architecture, data structures and information hiding,
- Factors that affect effective modular design: functional independence, cohesion, and coupling,
- User interface design,
- Software testing techniques: white-case, white-box, basis path testing (flow graph notation, cyclomatic complexity, graph matrices, etc.) black-box testing and system testing,
- Software testing strategies: unit testing, integration testing, validation testing and system testing,
- Software maintenance and configuration management,
- Cleanroom approach to software engineering,
- Reverse engineering and reengineering,

Assignments/Projects/Laboratory Projects/Homework

Group software development project (11 weeks)

Project milestones:

Project Plan Due	9/14/06
Requirements Specification Due	9/26/06
High Level Design Due	10/10/06
Detailed Design Due	10/24/06
Walkthrough	10/31/06, 11/02/06
Test Plan Due	11/9/06
Software, User Guide, Revised Documentation Due	11/21/06
Project Demonstrations	11/28/06, 11/30/06

Research paper (12 weeks)

In the course, all of you will be required to submit a software systems research paper, along the lines of a bibliographic essay, on a topic related to current methodologies, research ongoing in the field, and/or new applications. Sample paper topics, all specifically related to software development, include:

Math and Fundamentals:

Data Structures:

Architectural patterns and data structures for software development. Evaluation and selection of data structure(s): 6 hours

Algorithms and Software:

Software Analysis and Design - Overview of the analysis and design processes, best practices, techniques, documentation, object oriented analysis and design, UML. 15 hours

Problem Analysis – 4.5 hours

Solution Design – 6 hours

Computer Organization and Architecture:

Concepts of Programming Languages:

The selection of a language(s) for a problem. Specific languages and frameworks for requirements and testing. 6 hours

Social and Ethical Issues:

List the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e. g., test questions, essays, oral presentations, and so forth)?

Oral Communication (presentations):

to make 2 oral

presentations of typically 20 minutes duration. Include only material that is graded for grammar, spelling, style and so forth, as well as for technical content, completeness and accuracy.

All sections share the experience of planning, analyzing, designing, building, testing and presenting a software system as a self-organizing group.

Written Communication:

Every student is required to submit at least 1 written reports (not including exams, tests, quizzes, or commented programs) of typically 10 pages

Course Coordinator: Dr. Doris Carver

Last Modified: May 15, 2008