COURSE DESCRIPTION

Dept., Number  CSC 191  Course Title  Senior Project: Part II
Semester hours  2  Course Coordinator  Robert Buckley
URL (if any):  http://gaia.ecs.csus.edu/~buckley/

Catalog Description

Continuation of the group project begun in CSC 190. Teams apply software engineering principles to the design, implementation and testing of their software product. All technical work is published using guidelines modeled after IEEE documentation standards along with an appropriate user manual. Oral and written reports are required. Senior project is completed with the successful delivery, installation and demonstration of the software along with all approved documentation. Lecture one hour, laboratory three hours. Prerequisite: At least a C- grade in CSC 190 and full CSC or MATH/CSC major status.

Textbook

No textbook required.

References


Course Goals

Upon successful completion of this course, students and their teams should have demonstrated the following:

1. Worked effectively with one another, sharing the workload and responsibilities.
2. Effectively managed by the team’s project manager.
3. Met at least once weekly to review progress, to update the schedule and work plan. Time on each work assignment was collected, new assignments were made and meeting minutes were kept.
4. Met weekly with their faculty adviser, provided accurate status on work in progress and the team’s schedule. Documents were reviewed and discussed, and all changes to the team’s work plan and schedule were discussed and approved by the project sponsor.
5. Development followed the team’s management plan, including the team’s work plan and schedule.
6. Implemented an effective quality assurance process. For example, all work products were formally and thoroughly reviewed and approved by the group prior to submission to the
faculty adviser.
7. Meetings with project sponsor/users were well-planned and effectively managed with results and follow-up well documented.
8. The project sponsor was informed about project status and progress throughout the development of the software product.
9. All documents were approved by the team’s faculty adviser prior to conveyance to the project sponsor.
10. All changes to baseline products were processed according to the team’s documented change control process.
11. The completed set of product documents provided traceable “threads” from each specified requirement, to its design element, to its corresponding product component, to the system test case used to validate the implementation of the requirement.
12. All team members demonstrated an understanding of the software development process and the appropriate application of software engineering principles.
13. Analyzed testing needs to insure that the delivered system satisfies the requirements as specified in CSc 190.
14. Written a test plan, test specifications, and test cases.
15. Designed a software product from a requirements specification using state-of-the-art design tools and techniques and documenting that design in a design specification.
16. Implemented the software system from the design specification.
17. Tested the completed system in accordance with the developed test plan, test specifications, and test cases, and documented that testing in a test report.
18. Installed, demonstrated and delivered the software product and obtained user's acceptance of the delivered product.
19. Scheduled a final technical review – with the project sponsor – prior to the end of the semester. The team reviewed the work completed during the semester, discussed the system test results and installed and demonstrated the product. The Software Maintenance Manual CD (SMM-CD) was delivered. The project sponsor signed an acceptance and approval form, signifying formal acceptance of the product.

Prerequisites by Topic

Thorough understanding of:
- Software engineering principles and lifecycles: management and process.
- Technical writing and technical presentations.
- Overview of management functions.
- Introduction to project management.
- Project initiation (including cost estimation).
- Planning: tasks, costs, and schedules.
- Policies, strategies, rules, and procedures.
- Organizing: line and staff organizations, project, functional, and matrix organizations, project team structure.
- Staffing and training.
- Directing, motivating, leading.
- Controlling a project–including configuration management, quality assurance, reviews.
• Quality assurance using formal “walkthroughs” and “inspections.”
• Completing a proposal, planning, and requirements analysis (including user's manual) for a real world computer software/hardware project.
• Designing test specifications and test data.
• Designing a computer software/hardware system from requirements.
• Implementing a system from the design.
• Testing the completed system.
• Providing for appropriate user documentation.

Basic understanding of:
• Software engineering principles and lifecycles: management and process.
• Technical writing and technical presentations.
• Overview of management functions.
• Introduction to project management.
• Project initiation (including cost estimation).
• Planning: tasks, costs, and schedules.
• Policies, strategies, rules, and procedures.
• Organizing: line and staff organizations, project, functional, and matrix organizations, project team structure.
• Staffing and training.
• Directing, motivating, leading.
• Controlling a project— including configuration management, quality assurance, reviews.
• Quality assurance using formal “walkthroughs” and “inspections”.
• Completing a proposal, planning, and requirements analysis (including user's manual) for a real world computer software/hardware project.
• Designing test specifications and test data.
• Designing a computer software/hardware system from requirements.
• Implementing a system from the design.
• Testing the completed system.
• Providing for appropriate user documentation.

Exposure to:
• Software engineering principles and lifecycles: management and process.
• Technical writing and technical presentations.
• Overview of management functions.
• Introduction to project management.
• Project initiation (including cost estimation).
• Planning: tasks, costs, and schedules.
• Policies, strategies, rules, and procedures.
• Organizing: line and staff organizations, project, functional, and matrix organizations, project team structure.
• Staffing and training.
• Directing, motivating, leading.
• Controlling a project—including configuration management, quality assurance, reviews.
• Quality assurance using formal “walkthroughs” and “inspections.”
• Completing a proposal, planning, and requirements analysis (including user's manual) for a real world computer software/hardware project.
• Designing test specifications and test data.
• Designing a computer software/hardware system from requirements.
• Implementing a system from the design.
• Testing the completed system.
• Providing for appropriate user documentation.

Major Topics Covered in the Course

1. Designing the user interface.
2. Team dynamics.
3. Project development, management, and control.
4. Architectural and detailed design.
5. Design representation using UML.
6. Planning for system testing.
7. Developing system test requirements.
9. Applications and practices in government and industry.

Laboratory Projects

1. Prepare a software design specification document (9 weeks).
2. Prepare a system test plan (9 weeks).
3. Prepare a system test design and procedural test specifications document (4 weeks).
4. Prepare a system test report (2 weeks).
5. Prepare a users’ manual (2 weeks).

Estimated Curriculum Category Content (Semester hours)

<table>
<thead>
<tr>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Design</td>
<td>0.25</td>
<td>0.5</td>
</tr>
<tr>
<td>Comp. Arch.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prog. Languages</td>
</tr>
<tr>
<td>Data Structures</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Oral and Written Communications

Each student team is required to prepare a minimum of four (4) technical documents in which all team members are involved in the writing, reviewing and revising of each document. Documents must be approved by the team’s faculty adviser. Each student team is required to make two technical presentations in which all team members share equally in the presentations. The first presentation is made in class and typically is 15 to 20 minutes in length. The second presentation is made to the team’s sponsor.
Social and Ethical Issues

No significant component although the social and ethical issues are discussed using case studies that represent current examples of software development problems and the difficulties associated with various software products.

Theoretical Content

No significant component.

Problem Analysis

No significant component, although student teams are required to reexamine their understanding of a project’s requirements throughout the course.

Solution Design

A significant part of the CSC 191 course involves the student teams in the design of the software product they are developing.

/aa