SIMON FRASER UNIVERSITY
Senate Committee for Undergraduate Studies
NEW COURSE PROPOSAL

Course Number: CMPT 373-3

Course Title: Software Development Methods
Short Course Title: Software Development Methods

Course vector: 3 lecture

Course Description (for Calendar). Attach a course outline to this proposal.

Survey of modern software development methodology. Several software-development process models will be examined, as will the general principles behind such models. Provides experience with different programming paradigms and their advantages and disadvantages during software development.

Prerequisite: CMPT 276 or 275.

Corequisite: none

Course(s) to be dropped if this course is approved: none

Rationale for Introduction of this Course:

This course will be required in the proposed Software Systems program for the Surrey campus.

Scheduling and Registration Information:

Indicate effective **semester/year** course would be first offered and planned **frequency** of offering thereafter.

Fall 2009, initially offered twice annually in Surrey

Waiver required: no

Will this be a required or elective course in the curriculum?

Required in the Software Systems program.

What is the probable enrolment when offered?

40 students.

Which of your present CFL faculty have the expertise to offer this course?
Toby Donaldson, John Edgar, Anne Lavergne, Uwe Glaesser, Dirk Beyer, Rob Cameron, Janice Regan

Are there any proposed student fees associated with this course other than tuition fees? (if so, attach mandatory supplementary fee approval form)
no

Resource Implications:
Note: Senate has approved (S.93-11) that no new course should be approved by Senate until funding has been committed for necessary library materials. Each new course proposal must be accompanied by a library report and, if appropriate, confirmation that funding arrangements have been addressed.

Campus where course will be taught:
Surrey.

Library report status
____

Provide details on how existing instructional resources will be redistributed to accommodate this new course. For instance, will another course be eliminated or will the frequency of offering of other courses be reduced; are there changes in pedagogical style or class sizes that allow for this additional course offering?
See attached Software Systems Curriculum document.

Any outstanding resource issues to be addressed prior to implementation: space, laboratory equipment, etc.
See attached Software Systems Curriculum document.

Approvals

1. **Departmental approval** indicates that the Department has approved the content of the course, and has consulted with other Departments and Faculties regarding proposed course content and overlap issues.

   ____
   Chair, Dept./School Date

   ____
   Chair, Faculty Curriculum Committee Date

2. **Faculty approval** indicates that all the necessary course content and overlap concerns have been resolved, and that the Faculty/Department commits to providing the required Library funds.

______________________________ Date: ____________________
Dean or Designate

List which other Departments and Faculties have been consulted regarding the proposed course content including overlap issues. *Attach documentary evidence of responses.*

____
____
____
____

Other Faculties approval indicates that the Dean(s) or designate of other Faculties affected by the proposed new course support(s) the approval of the new course.

______________________________ Date: ____________________

______________________________ Date: ____________________

3. SCUS approval indicates that the course has been approved for implementation subject, where appropriate, to financial issues being addressed.

Course approved by SCUS (Chair of SCUS)

______________________________ Date: ____________________

Approval is signified by date and appropriate signature.

Proposed CMPT 373 Course Outline

This course exposes students to modern software development methodology. Several software-development process models will be examined, as will the general principles behind such models. Students will gain experience with different programming paradigms and their advantages and disadvantages during software development.

Topics:
- Software development process models: component-based development, iterative processes
- Agile software development: extreme programming, test-driven development
- Programming paradigms for special purpose software: scripting, relational, prototyping
- General paradigms: Object-oriented, functional, and logic programming
- Reengineering, reverse engineering
- Design patterns, refactoring, aspect-oriented programming
- Formal development methods (formal requirement specification, cleanroom, OCL, Z)

Grading:
Assignments 75%, final exam 25%

Textbook:
none. Web-based resources will be used.