

**Course number and name:** **CS 04321: Software Engineering I**  
**Credits and contact hours:** 3 credits / 3 contact hours  
**Course Coordinator:** Jack Myers  
**Instructional Materials:** Software Engineering, Ian Sommerville, 2016.

### Specific course information

**Catalog description:** An introduction to the discipline of Software Engineering. Students will explore the major phases of the Software Lifecycle, including analysis, specification, design, implementation, and testing. Techniques for creating documentation and using software development tools will be presented. Students will gain experience in these areas by working in teams to develop a software system. Proficiency in programming is expected of the students entering this course.

**Prerequisites:** (CS04222 Data Structures and Algorithms or CS04225 Principles of Data Structures) **and** (COMP 01112 College Composition II or HONR 01112 College Composition II Media Literacy or ENGR 01201 Sophomore Engineering Clinic I) **and**(CMS 06202 Public Speaking or ENGR 01202 Sophomore Engineering Clinic II)

**Type of Course:**  Required       Elective       Selected Elective

### Educational objectives for the course

1. **Agile/Scrum.** Students have applied the Agile/Scrum methodology to a development project. This includes fully adhering to Scrum values, assuming Scrum roles, participating in Scrum events and producing Scrum artifacts.
  - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
  - ABET (5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
2. **collaborative development.** Students have worked collaboratively in a team to implement and fully test detailed designs and code.
  - ABET (5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.
3. **project initiation.** Students have demonstrated an understanding of the key elements of project initiation.
  - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

4. **requirements documentation.** Students have translated end-user requirements into system and software requirements, in a structured software Requirements Document that combines user requirements and functional requirements.
  - ABET (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
  - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
  - ABET (3) Communicate effectively in a variety of professional contexts.
5. **requirements elucidation.** Students have collected and analyzed user requirements using formalisms such as User Stories, UML, and business process modeling.
  - ABET (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
  - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
6. **system design and implementation.** Students have identified and applied appropriate software architectures and patterns to carry out high level design of a system. Students have critically compared alternative choices and structured this information in an architectural Design Document. Students have implemented an application as specified in the Design Document.
  - ABET (1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
  - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
  - ABET (3) Communicate effectively in a variety of professional contexts.
  - ABET (6) Apply computer science theory and software development fundamentals to produce computing-based solutions.
7. **technical presentations.** Students have reflected on the outcome of a project and presented their reflections in a verbal report.
  - ABET (3) Communicate effectively in a variety of professional contexts.
8. **testing methodology.** Students understand how to produce comprehensive test plans and scripts.
  - ABET (2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
  - ABET (3) Communicate effectively in a variety of professional contexts.

### **Required list of topics to be covered**

1. Software processes
2. Agile software development and scrum

3. Requirements engineering
4. Design documentation and processes
5. Architecture documentation
6. Software testing and maintenance
7. Utilizing GIT

Optional list of topics that could be covered

1. Formal validation: IQ, OQ and PQ
2. Project management using Microsoft Project
3. UML diagramming tools
4. Creating physical database diagrams
5. RESTful web services
6. ITIL