Course Syllabus
CSCI 3308: Software Development Methods and Tools

1. Course number and name
CSCI 3308 Software Development Methods and Tools

2. Credits and contact hours
3 credits, two lectures per week (50 minutes each) and 1 2-hour lab per week (110 minutes)

3. Instructor's or course coordinator's name
Elizabeth Boese

4. Text book, title, author, and year
Practical Software Development Techniques (Fourth Edition), Edward Crookshanks, 2013
a. other supplemental materials
Unix, Regex Golf, Regex (http://bit.ly/1vGFwJV), Makefiles, [optional] Ant, Bash Shell Scripting, Software Process & Methodologies by David Kung, Scrum Framework (just the chapter on Scrum Framework), Version Control by Erik Sink Chapters 1, 2, 5, 6, 8, Planning Poker, RDBM, E-R Diagrams, Intro to SQL Pages 1-13 (stop before views) and 16 group by/having, NoSQL, XML tutorial, JSON tutorial, SOAP vs REST, Retrospectives, Cloud, Testing, Klocwork, Find Bugs

5. Specific course information
a. brief description of the content of the course (catalog description)
Covers tools and practices for software development with a strong focus on best practices used in industry and professional development, such as agile methodologies, pair-programming and test-driven design. Students develop web services and applications while learning these methods and tools.
b. prerequisites or co-requisites
CSCI 2270 Computer Science 2: Data Structures
c. indicate whether a required, elective, or selected elective (as per Table 5-1) course in the program
required

6. Specific goals for the course
a. Specific outcomes of instruction.
The student will learn the fundamentals of software development methods and tools you are likely to find in use in the workplace.

The student will acquire state-of-the-art skills that will not only help them do their work in other programming classes but will also give them a very useful vocabulary to use on job applications and during interviews.

The student will be able to use a variety of tools introduced in class.

The student will be able to choose the best tool for use in a specific situation.

The student will be able apply what you learn in the context of a small group semester long project.

b. explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.

a) Apply Knowledge
b) Computing Requirements
c) Design System
d) Teamwork
f) Communicate Effectively
g) Analyze Impacts
h) Professional Development
i) Current Techniques
j) Design Tradeoffs
k) Design & Development

7. Brief list of topics to be covered

- Software Process Models
- Waterfall, Iterfall, Agile
- Software Patterns
- Architectural Design
- Requirements Elicitation and Analysis
- Documentation
- Licensing/Copyright/Patents/IP
- Code Quality
- Testing Methods and Strategies
- Static and Dynamic Analysis
- Code Repair Strategies
- Refactoring
- Code/Peer Review
- Code Management
- Source Code Control and Managing Conflicts
- Software Configuration Management Systems
- Bug Tracking
- Managing Dependencies/Build tools/Integration
- Continuous Integration
- Unix tools
- Data stores/access/formats/models
- Software Deployment and Deployment environments/models: Development Environments