CSCI E-599 Software Engineering Syllabus (rev. Jan. 28, 2016)
(Subject to Revision)

Course Overview:

The capstone CSCI E-599 course is intended for more advanced software developers and serves as an alternative path to the thesis in the software engineering concentration. The core of the course consists of a large semester-long group project that will include requirements gathering, forecasting, designing, coding, testing, and documenting. The accompanying lectures aim to provide timely concepts from the software engineering body of knowledge as they relate to the course project. We will also have class discussions and demonstrations around practical aspects of improving your software-related skills that draw upon the students' collective experience and upon the research. The course aims to make students better versed in the field, to improve awareness of the standard tools and techniques, and to become more capable team members/leaders in software development projects.

Prerequisites:

Students should be degree candidates in the Graduate Program in Information Technology. They should have completed nine courses (including CSCI E-97) in the software engineering concentration and have proficiency in Java, or permission of the instructor.

Instructors:

Peter V. Henstock, Ph.D.
capstoneprof@gmail.com = individual email for questions/comments
cscie599@gmail.com = course email: please CC on group communications

Teaching Fellow: Robert Zupko
rzupko@mtu.edu

Class Meetings:

All class meetings will occur online only. There is no physical classroom to attend lectures in person.
Weekly 7:40-9:40 pm EST Thursday starting Jan 28
Preferred live: 7:40 – 8:40 pm EST Thursday lecture
Required live: 8:40 – 9:40 pm EST Thursday discussion
Required live: Attend 3 milestone meetings 7:40-9:40 (see Important Dates)
Required live: Final presentation for faculty panel (date TBD)

Required live indicates your presence is required online at the specified time with a connected computer microphone. If this presents a difficulty, please contact the professor.

Required Textbook: None. Research articles will be used for the discussions.

Grading:

The course grade will be based on a combination of six criteria to capture both group and individual efforts:

36%	group project divided into 3 milestones and final presentation to faculty committee; each with a presentation and report
20%	assessment of the contribution to the group project by each group member
8%	short homework assignment in the first weeks of the course
6%	software engineering productivity tool presentation
10%	discussion led by each student on a software engineering research topic
10%	active participation in course discussions and course
10%	short paper describing your active self-improvement in software engineering

Lecture Format:

To accommodate the generally diverse geographic distribution of students, lectures will be held for two hours online from 7:40-9:40 pm EST using provided web-based software.

The course meetings will be roughly divided into two parts each week. The first portion will be mostly lectures on key topics from the extensive field that aim to tie into your project such as requirements gathering, debugging strategies, testing, project management, user interfaces, etc.

The second portion is intended to be more interactive during which we will discuss ideas and opinions on best practices in software engineering such as the popular software engineering software, the practice of writing unit tests, writing clean code, and managing projects. To facilitate the discussion aspect, we expect students to attend the second hour online live from 8:40 to 9:40 pm EST Thursday. If this presents a difficulty, please contact the professor.

Software Engineering Productivity Tool Presentations:

As the industry changes, an ever increasing number of software tools aim to increase the productivity of software engineers. Each student will learn about and give a short demonstration on what it does, why it's useful, and how to get started. These presentations will be held early in the semester so the appropriate tools can be part of the project workflows.

Discussions on Software Engineering Topics:

The Thursday night discussions will typically bed led by one (or two) students. Each discussion will address one research question of interest posed by a student and we will collectively explore answers from both academic and industry perspectives. The discussion leader will post the question and a citation or link to a supporting research article to the discussion board one week prior. The rest of the class will prepare to participate in the discussion by finding a different relevant article and posting the article and a few line summary to the same discussion board before class.

There are three motivations for this. First, the popular literature offers many vague recommendations on how to perform software engineering tasks. However, there is also a lot of research that is assessing and defining best practices. While this is not a research class, one goal is to be able to draw upon these resources. Second, we have students working in different areas of software development. The discussions aim to tie in the research to how different organizations function in practice. Third, the software engineering field is quite wide and this allows students to engage and dive deeper into topics of their interest.

We will ask for topic plans after the first few weeks of class and schedule each student to lead one discussion to coincide with lecture topics.

Milestone Meetings:

The main project of the course has three milestones that lead up to a presentation before the Harvard Extension Information Technology faculty. Each milestone will include a report and a presentation that will be followed by Q&A and discussion. Live attendance at the lecture and discussion is required.

Software Engineering Management:

A theme throughout the course will be on managing teams for software development. The final project paper will be a perspective on an aspect of software engineering for teams that relates at least somewhat to software engineering management. Part of the paper will be an analysis of how you would lead the team given a second chance.

Planned Lecture Topics:

Introduction to Software Engineering. Problems and challenges.
Requirements gathering
Estimating project time
Software methodologies
Project management and software teams
Software metrics
Testing: unit testing, random testing, integration testing, etc.
Debugging: formal strategies
User interface design
Software reuse strategies

Important Dates (subject to change):

Jan 28	First Lecture	
Feb 4 or 11	Project rollout	
March 3	Milestone 1:	Project proposal, estimated effort, initial design Teams will submit a report and give a presentation
March 14-18	Spring Break.	No lecture. Project work continues
April 7	Milestone 2:	Progress report Teams will submit a report and give a presentation
May 5	Milestone 3:	Teams will submit a report and give a presentation
May 5	Final class	
~May 12	Presentation to faculty committee. Final report due 4 days prior.	