

Class Notes

Suggested Topics

Exam II

Selfi

Presentation

CSCI 565: Advanced Software Engineering

Instructor : Dr. Hassan Reza
Email : Hassan.reza at engr. und.edu
Phone: 701.777.4127
Office hours : 1:00 to 2:15, T, TH or by appointment
Class meets : 4:00-5:15p, Rm 164, Upson II
Office : 366E, Upson II
Credit hours : 3

Textbooks:

- **Referenced Text:**

- Practical Model-Based Testing, by Mark Uttung and Bruno Legeard
- The Art of Software Testing, G. Myers, 2nd edition, Wiley

- **Required Text:**

- Software Testing: A Craftsman's Approach by Paul C. Jorgensen, 4th edition, 2014.

Other Sources:

- ACM/IEEE Journals and Proceedings Articles

On-line Materials Via UND Main Library:

Course lecture notes will be available on-line from the course web site. For the most parts, I will do my best to provide handouts and copies of some of the reading relevant to this course. However, this is an advanced course and by its very nature requires frequent visits to the library (UND's Chester Fritz library and/or NDSU, elsewhere) resources. Therefore, it is YOUR responsibility to get copies of journal articles and bring them to the class or to use them for homework, project, or term paper. Currently, both ACM and IEEE are on-line via UND main library. Please consult with Chester Fritz library people to find out how to connect to On-Campus and Off-Campus from your computer in order to access on-line IEEE/ACM digital libraries.

Course Description:

Software testing is a process by which software systems are executed in order to determine whether the product is built right (verification) or the right product has been built

(validation). The past decades have witnessed a great increase in the study and hence in the theory of software testing techniques. This course covers the most well-known testing methods/strategies, such as structured testing, functional testing, mutation testing, data flow testing, and formal specification based testing methods. Upon successful completion of the course, the student should be able to understand the theory and the role of software testing and its relationship to the quality assurance. The objectives for this course include but not limited to:

- Basic Definitions
- Set theory and graphs
- Software Testing Strategies
- Functional testing
 - Boundary value analysis
 - Equivalence classes
 - Decision tables
- Structural Testing
 - base -paths
 - McCabe's Basis Path Method
 - Essential Complexity
 - Define/Use Testing
 - Data Flow testing
 - Slice-based testing
- Integration and System Testing
 - Recovery Testing
 - Performance/Stress Testing
 - Regression Testing
 - Alpha/Beta Testing
- Nonfunctional Testing
 - Security Testing
 - Usability Testing (web)
 - Reusability Testing
- Hybrid testing (Integrated functional/structural testing techniques)
- Advanced Testing Techniques
 - Testing Software Architectural Design
 - Portability
 - Deployability
 - Usability
 -
 - Model-Based Testing of :
 - Safety Critical Systems
 - Security Critical Systems
 - Cyber Physical Systems
 - Cloud Computing

- Testing High Performance Computing Systems

Course Prerequisites and Prerequisite Fulfillment:

The course will utilize lecture and discussion, as the main teaching tools for presenting the course material. Students will be expected to read the material be prepared to discuss the readings and the assignments in class.

Prerequisite for this course is successful completion of Csci463, background in discrete math, and exposure to any high level programming languages such as C, C++, or Java are required. This course is a CORE course for graduate students and hence demands both independent and cooperative work. Working independently requires that you have good time management skills and self-directed learning skills. There are no prerequisites for team work other than a willingness and flexibility to participate and learn as part of a team. This course fulfills all or part of the prerequisite requirements for CSci 566 (Software Engineering Track). This course is also very relevant for students with a general background in computer science, who are interested in the theory and practice of software development and maintenance.

Other Related Courses in Software Engineering Track (SET):

- **Software Architecture** (CSci 582):
 - A study of formal and semi-formal notations used to specify communicational and computational elements of a software system;
- **Formal Specification Methods** (CSci 562):
 - This course is an introduction to the application of modeling notations that allows a software engineer to precisely specify and verify the behavior of a system. In this course, we will discuss the suitability of various specification techniques.

Email Policy:

You are more than welcome to email me about your concerns at any time. If you have any specific question about the course that requires some lengthy discussion then you better off to see me at the office.

Exams:

The exams will contain questions covering material in the class, homework, the lecture notes, textbook, and handouts. There are two exams:

- **Exam 1: Nov 7th**
- **Exam 2: Dec. 11th**

Make-up Exam and Quiz Policy:

No make-up exams or quizzes will be conducted.

Assignments and Pop Quizzes:

There will be several assignments and quizzes throughout the course. The topics of these assignments will be taken from reading resources and will be announced in advance. Assignments should be turned in the class. For every day of late submission 20% of the points will be deducted. You are advised to hand in assignments in class. Any homework placed without my knowledge (e.g., in my office mail box) will be accepted without any warranty. In case an assignment is lost, it is your responsibility. Recommendations are: to save everything, retain important papers and computer files until the course is finished.

There will also be pop-quizzes on material we have recently read/discussed in class. The quiz will normally be one or two basic questions. Your answer will be a given S (**Satisfactory**) or U (**Unsatisfactory**). There will be 6-8 quizzes for the course. To accommodate those who miss class, two low quizzes for each student will be dropped.

Submitting Assignments:

All programming assignments (if any) must be submitted either by email or on the disks/CDs.

PLEASE DO NOT submit/send any ZIP FILE.

The following guidelines are used as the grading criteria:

1. If the programming assignment cannot be compiled due to compilation errors, the assignment will receive a grade **F**;
2. If the programming assignment compiles with warnings, the assignment may lose up to 100% of your grade depending on the nature and the severity of the warnings;
3. Your program must adhere to proper documentation and programming styles (e.g., comments);
4. Your submission must include both the source code and executables file together with the samples of output(s);
5. If the assignment is submitted by email, you need to send your submission as attachments as follows:
 1. Include your name, assignment number, and class in the subject heading (e.g., **Csci565.John-Smith.HW1.**);
6. If the assignment is submitted on a disk/CD, use a label with your name, assignment number, and class on it.

Note 1: All non-Programming assignments must be typed and spell checked otherwise they won't be graded.

I normally return assignments, exams, and quizzes to the class. If you are absent, it is your responsibility to pick up your work at my office.

Term Paper:

There is one term paper which counts 45 % of the course grade. The main objective will be to identify a killer idea to extent or investigate in detail a specific issue related to the **software testing** that attracts you. This may include all papers and technical reports on the subject as well as personal communications with the lead investigators. The term paper consists of three parts as follows:

(1) Topic selection and team organization. Each team consists of TWO students. You need to identify your research topic and a team member whom you can collaborate to write your research paper. The topic title and a brief description of the research (one page) together with team members must be submitted in MS-DOC by **September 6, 2018**

(2) Survey Report. Once a team organized and research topic is selected, your team will perform literature research on the topic, write a high quality paper on it. The length of the report should be 10-12 pages white-paper 10 point font, single space, *IEEE/ACM* format. The survey part must describe the problem domain, with proper problem statement, and a survey and analysis of essential work in that specific area. The survey part in MS-DOC is due on **October 25, 2018**

(3) The research part should be a *publishable* paper (8-10 pages). The research part **MUST** describe your own attempt to either solve a problem in this domain or go a long way towards its solution. Your report **MUST** be readable and fully **cited in the reference section** at the end and number. Each paper should be referred to in the context section by that number and there **MUST** be brief discussion or summaries of contribution of the paper and a comparison of its results with the contribution of the other papers. The context section should contain ALL fundamentally important papers in the topic area right up to the present. This may include all technical reports/papers recently published by the important contributors. The full term paper in **MS-DOC** must be in by **December 13.**

The recommendation is that you start thinking or working on term paper ASAP.

Presentation:

Each team will be assigned to make a 20-25 minutes of PowerPoint presentation to show their results. Allow 30 minutes for presentation and 5 minutes for answering questions. Both team members will give a report on their part of the project in the designated 20-25 minutes for the team. The report will be scheduled alphabetically by the last name of team members. The presentation will begin from **December 4th through December 7th**.

Course Grading Policy:

The course grade will be computed as a combination of the following factors:

- Two Exams (40 %)
- Pop Quizzes (5 %)
- Homework (5 %)
- Term papers (50 %) consists of:
 - Research Topic selection and Team organization (2%)
 - Survey (20%)
 - Research papers (25%)
 - PowerPoint Presentation (3 %)

Grading Scale:

90 - 100%	A
80 - 89%	B
70 - 79%	C
60 - 69%	D
below 59%	F

Participation & Attendance Policy:

Regular classroom attendance is required (this includes any team meeting held class time and outside of class time), but does not count as participation. Active participation in the class involves preparing ahead of time, asking intelligent questions that the book does not answer, answering questions in class, and participating in classroom discussions. If you know you will miss a class ahead of time, email the instructor. Make friends with your classmates so they will be willing to share notes, which problems were assigned, etc., for when emergencies arise and you do miss class.

Civility in the Classroom:

Students are expected to assist in maintaining a classroom environment, which is conducive to learning. Please do not bring food to the classroom and always turn your cell phone (ringer) off if you have to have it with you.

Disability Conditions:

If you have a disability condition that may affect your participation in this course and wish to discuss academic accommodations, please inform me as soon as possible.

Drops & Withdrawals:

Students should not assume that the instructor will administratively drop the student for non-attendance. Withdrawal from the course is the responsibility of the student. Should a student decide to withdraw from the course, simply not attending class sessions, rather than formally withdrawing from the course, may result in an 'F' on the student's transcript.

PLAGIARISM, PROFESSIONAL ETHICS & PERSONAL INTEGRITY

*The UND academic policy clearly states an expectation that each student submit his/her own work do otherwise, that is, to submit another's efforts as one's own, is at its least plagiarism, and at its most unethical and without integrity. Indeed, it is a matter of professional ethics and personal integrity to graciously accept accolades for a job well done, as well as to publicly acknowledge those to whom partial credit is owed. Those found to be cheating on exams, assignments, and Project will receive an **F** for the COURSE, and the matter will be discussed with the appropriate Dean and/or Chair. I do not anticipate any problem here.*

*Note: It is absolutely unacceptable and unethical to re-submit a report/work that has already been submitted and received credit as part of the requirement for another course (e.g., Csci582, etc.). The team will receive grade an **F** for the re-submission of the existing work. Also, cut and paste of any published works, figures, tables, results, etc. by your report is considered unethical, and illegal. If detected, your TEAM will receive at least an F for the work or for the Course.*

Disclaimer: This syllabus is intended to suggest the outline of the course; it is not absolute. Changes to the syllabus, should they occur, will be announced in class.

Last Update August 20, 2018. by H.R.