CS777: Software Reliability and Quality Assurance
CRN 21804
Spring 2009

Synopsis

Software reliability is defined as the probability of failure-free operation of software for a specified time period within a given environment. This course motivates the need to engineer reliable software and focuses on approaches for measuring, assessing and improving reliability as part of a wider software quality assurance program. The course introduces the basic concepts of reliability (e.g., incidents, faults and failures), compares and contrasts software with hardware reliability, and examines a number of classic software reliability models. The course emphasizes techniques through which to assess the trustworthiness of reliability claims and predictions, and highlights a number of approaches to improve them. The course will also look at the role of measurement and testing in software reliability engineering, and explore the broader issue of software dependability (e.g., safety, availability and security). Software quality fundamentals and wider quality assurance activities will be examined throughout the course.

Credits

4

Prerequisites

A passing grade for CS615 and CS616 – the graduate Software Engineering sequence (or its equivalent). A passing grade for CS775 (Systems Requirements Engineering) is also recommended. Students without these prerequisites take this course at their own risk!

Instructor

Dr Olly Gotel

Email: ogotel@pace.edu
Website: http://csis.pace.edu/~ogotel/teaching/cs777spring09.htm
Office: Room 212, 163 William Street, New York
Office Hours: Monday 4:00pm-5:00pm; Tuesday & Thursday 2:30pm-4:30pm
(Any necessary changes to this schedule will be listed on my website, but always email me first if you are making a special trip)

Time & Place

Monday 5:00pm-8:35pm; Room 236, 2nd floor of 163 William Street, New York

Course Details

1. Aims

This is a rather ambitious set of aims! Students are encouraged to investigate tool support for the topics covered -- it may not be possible to examine these within the class sessions.

- To explore the techniques, value and costs associated with assuring and improving software quality;
- To examine how requirements define the required quality characteristics for a software product, and show how they influence the measurement methods and acceptance criteria that can be used for assessing these quality characteristics[1];
- To explain the fundamental concepts of software reliability engineering and clarify the distinction between the failure process of hardware and software;
- To study the reasons for, and problems associated with, assuring the reliability of software products;
- To show how to measure (and how not to measure) the reliability of software, highlighting the role and importance of measurement in software engineering;
- To propose how to put together a testing plan to collect operational failure data for reliability evaluation;
- To survey a number of software reliability models and to illustrate how they work in practice to assess reliability growth;
- To demonstrate how to critique the predictions of software reliability models and use recalibration techniques to improve their accuracy;
- To investigate the broader concerns surrounding software dependability[2] and explore alternative strategies for its evaluation and improvement;
- To provide initial experience in constructing a software quality assurance plan (focusing on dependability plans), in designing quality audits (process and product), and in exploring their effectiveness;
- To raise awareness of dominant industry initiatives in the area of software quality, such as ISO 9001, SEI CMMI, Total Quality Management (TQM), Quality Function Deployment (QFD) and Six Sigma, and to critique the role of certification and important auditing regulations.

2. Learning Outcomes

After completing the course, students will be able to:

- Describe what software quality is, why it is important, and know how to assess and monitor important quality attributes on a software development project;
- Understand the fundamentals of reliability engineering and explain the relationship between faults, failures and incidents;
- Discuss the particular engineering and practical issues associated with developing and assuring the reliability of software-intensive systems;
- Specify reliability requirements and understand the costs associated with both engineering these requirements and certifying their attainment;

[1] Note that reliability is one quality characteristic, and it is often one of the more critical characteristics to specify and demonstrate.

[2] Note that dependability is a broader quality characteristic that encompasses reliability.

- Put together a practical measurement scheme for assessing software reliability;
- Set up a pragmatic testing program to gather and analyse meaningful failure data, during both development and operations, in order to model and track reliability growth;
- Compare and contrast different approaches for measuring, predicting and improving software reliability;
- Apply software reliability modelling techniques to failure data to obtain reliability estimates and know they are accurate;
- Understand the meaning and value of third-party claims about software reliability (i.e., appreciate the strengths and limits to reliability modelling);
- Recognize the relationship between testing/debugging and reliability growth, in order to make informed decisions about the costs and benefits of testing programs;
- Determine whether a software product meets specified reliability requirements in order to make informed decisions about deployment;
- Explore the literature and practices associated with the broader issue of software dependability and software quality;
- Define, document and deploy an appropriate dependability/quality plan for a small software project;
- Understand the goals of a software quality assurance program, the basics of a statistical approach to quality control and the requirements for establishing a quality culture when engineering software.

3. Method of Instruction

This course will run as a tutorial. This means that the class will revolve around weekly and interactive group meetings/discussions, student presentations and involve guided self-study. The course will be supplemented by a number of guest presentations in a couple of specialist areas to focus on the Quality Assurance component of the course. Students will be expected to attend each scheduled class session and undertake (at least) 15 hours of coursework per week (as required for a 4 credit course).

Note that this course will NOT provide Powerpoint hand-outs to students. Students are expected to be active participants in this class and will be expected to take detailed notes during the class sessions.

4. Assessment & Grading

The course will be assessed through small weekly tasks set by the instructor, two papers on quality topics, and by final examination Students must pass each assessed component of the course to get a passing grade. Over 3 unexcused absences from scheduled class meetings will result in a reduction of grade for the semester.

Assessment Type	**Assessment Weighting**
Weekly homework, notebook, in-class presentations and participation (binary continuous assessment)	10%
Two term papers (and class presentation on each paper)	40% (20% each)
Final exam (cumulative)	50%

Grades will be based on the following scheme:

Letter grade	Level reached
A	Dominates the material
A-	Masters the material
B+	Good understanding, some excellent work
B	Good understanding
B-	Aptitude
F	Too weak for graduate work

Incompletes: in order to be fair to those students who complete the course in a timely manner, the grade of those students taking an incomplete will be reduced by a full letter grade (i.e. from A- to B-) for each semester, or portion thereof, that the incomplete is in effect.

4.1. Weekly Homework / Presentations / Participation

Students will generally be given a small weekly task to either explore the material covered in the day's class or to prepare for an upcoming class. This will involve a mixture of reading, written research work, practical work and in-class presentation. Detailed instructions will be given in each class and posted on a weekly basis on my website (see weekly notices at the bottom of the web page): **http://csis.pace.edu/~ogotel/teaching/cs777spring09.htm**. This component of the assessment will be graded in a binary fashion (i.e., pass/fail, 1/0, you either do it or you don't). Failure to complete any task will probably jeopardize your performance on the final exam! As graduates, you are expected to be responsible for your own work ethic.

In addition, this course will not provide powerpoint hand-outs to students. Students will be expected to take detailed notes during the class sessions. I will ask to see a student's class notes periodically during the semester.

4.2. Term Papers

Each student will prepare two term papers on specialist topics relating to the course (to be agreed with the instructor). The term papers are designed to provide students with the opportunity to pursue an aspect of Software Quality Assurance or broader Software Dependability (e.g. Safety Critical Systems, High Availability Systems, Secure Systems, etc.) that is not explicitly covered in the class sessions. Topics may include the role of quality certification, understanding Six Sigma, the cost of quality, navigating ISO Quality Standards, Techniques for Developing Reliable Software, such as Cleanroom Software Engineering or Model-Checking, Setting up quality management plan for a project, the role of software process improvement, etc.

The term paper is to be prepared in accordance to submission guidelines that will be provided once the course is in progress. Interim submission dates for draft feedback will also be provided and students will be expected to give a class presentation on their paper during scheduled sessions and to lead a guided discussion to get class feedback prior to final submission.

4.3. Final Exam

This class will also be assessed by a final three-hour exam. The final exam will be closed book and cumulative (i.e., everything will be examinable). Detailed instructions concerning the exam will be given towards the end of the semester and a past exam paper will be provided for guidance. See the schedule for dates.

5. Policies

5.1. Learning Contract

A key assumption of the course is that students must take responsibility for their own learning. While instructors can help select and steer students through a wealth of material, students themselves must take the lead and drive their own progress, particularly at graduate level. University life requires students to organise and make best use of their own time; it is the instructor's role to guide students, not to chase them. The classes will cover key concepts, techniques and methods as a starting point, but students must explore and practice these to master them.

5.2. Special Accommodation

If a student has a disability that may affect academic performance, that student should notify Pace University's Counselling Centre for assessment and advisement. It is encouraged that students who may need accommodation do this as soon as possible, since this can **only** be permitted with an approval letter from Pace Counselling Services. Please note, accommodation will not be retrospective; it will only take effect from receipt of an approval letter.

5.3. Plagiarism

You are encouraged to work with others, help each other with problems, communicate ideas, talk through work, exchange tips – it is an important part of the learning process. However, blatantly copying another student's work or work you have found via other sources, submitting it under your own name and not acknowledging who did the work is cheating. Cheating is not tolerated and will be dealt with according to CSIS policy (please see my website for a copy of this policy).

6. Preliminary Timetable & Content

This timetable and suggested content is provided for guidance. It is not fixed and the instructor **reserves the right to change this at any time**, if considered appropriate. Students are expected to check the schedule on the class web site regularly for any changes and updates: http://csis.pace.edu/~ogotel/teaching/cs777spring09.htm. It mostly covers the reliability topics at the moment since this will be the theme of the instructor-led discussions. The Software Quality Foundations and Quality Assurance topics will be based on the student term paper choices and will be student/external guest-led. You will have a number of guests and their visit dates are to be determined. Their presentations will cover: useful quality techniques and tools; measuring value on projects – the role of quality; and assuring quality on agile projects. Most classes will be organised to cover both the reliability and quality themes.

TENTATIVE

Session	Monday	Topic
1	01/26/09	Individual preparatory course work
2	02/02/09	Overview of course objectives and assessment Why Measure the Reliability of Software? What is the Association with Quality Assurance?
3	02/09/09	Forensics – Picking Apart Disasters to Learn What is Sufficient Quality/Reliability?
	02/16/09	*No class – Presidents' Day*
4	02/23/09	The Failure Process and the Reliability of Hardware and Software Software Reliability Modelling – Basic Concepts and Models
5	03/02/09	Software Reliability Modelling – Predictive Accuracy and Recalibration of Models Understanding the Limits to Reliability Evaluation and Ultra-High Assurance
6	03/09/09	Presentation of term papers – QA topics
	03/16/09	*No classes – Spring Break*
7	03/23/09	Term paper 1 due Practical Exploration of Failure Data and Reliability Models (using tools)
8	03/30/09	Measurement in Software Engineering and the Fundamentals of Measurement Theory (and Software Metrics) Software Reliability Data Requirements (QA Data Requirements)
9	04/06/09	The Measurement of Incidents, Faults and Failures The Measurement of Execution Time Quantitative / Qualitative Measures of Quality
10	04/13/09	Data Collection Forms and Procedures for Reliability Assessment Data Models and Database Structures for Reliability Assessment Testing and Defect Reporting/Tracking
11	04/20/09	The Role of Software Testing in Quality Assurance Types of Testing; Testing Plans Knowing You are Done!
12	04/27/09	Presentation of term papers – QA topics
13	05/04/09	Term paper 2 due Revision class / practical exercises putting things together
14	05/11/09	FINAL EXAM

7. Resources

The course will not be following the chapters of a single textbook. Instead, it will draw upon the research and practice of the Centre for Software Reliability (CSR), City University, London. CSR is one of the world's major players in dependability research and industrial consultancy. I would therefore like to acknowledge the kind assistance of Pete Mellor in my prepartion of the reliability aspects of this course.

It is recommended that students explore the following FREE online resource as a complement to the class sessions:

- Lyu, Michael R. (Editor in Chief), Handbook of Software Reliability Engineering, McGraw-Hill, 1996.
  [available from: http://www.cse.cuhk.edu.hk/~lyu/book/reliability/]

Other key, though optional, resources that are recommended for this course include:

- Fenton, Norman. E. and Pfleeger, Shari. L., Software Metrics: A Rigorous and Practical Approach, 2nd Edition, PWS, International Thomson Computer Press, 1998.
- Musa, John D., Software Reliability Engineering: More Reliable Software Faster and Cheaper, 2nd Edition, Author House, 2004.
- Pham, Hoang, Software Reliability, Springer-Verlag, 2000.

NO NEED TO BUY!

Links to additional resources (there is an abundance of free material on Software Quality Assurance) will be posted on the course website as the course progresses and when particular topics and papers are relevant – else I will overwhelm you! Students may also find it useful to refer to their CS615 and CS616 general Software Engineering texts and CS777 Systems Requirements Engineering material.

Dr Olly Gotel
Assistant Professor
Department of Computer Science
CSIS, Pace University
ogotel@pace.edu; http://csis.pace.edu/~ogotel
Spring 2009