CSE6340: Advanced Software Engineering

Prof. Jeff Tian
E-mail: tian@seas.smu.edu
Home page: http://www.seas.smu.edu/ tian
CSE, SMU, Dallas, TX 75275
(214) 768-2861; Fax (214) 768-3085

Class:	TTh 5:00–6:20pm at 214C
Office hours:	TTh 2:00–4:00pm, SIC 318
Course material:	Debra McDowell (214) 768-3080
	debramcd@seas.smu.edu

1. General Description

In this advanced software engineering course, we will have in-depth discussions about several important and inter-related topics concerning the measurement, analysis and improvement of software processes and products. The major focus will be on three areas: 1) software reliability engineering and recent development; 2) software measurement and statistical analysis; 3) formal models for software metrics evaluation and selection. About half of the class time will be spent on topic 1 above, and the rest distributed about evenly among the rest two.

* Notice that the above description is somewhat different from the generic description for CSE 6340 in your "SMU Graduate Bulletin". We are doing a in-depth study of several important topics in software engineering instead of a superfacial survey of all the topics.

2. Textbooks and Class Material

There is no required textbook. However, the student are required to obtain a copy of the collection of research and application papers related to the above three topics that I have compiled. Additional papers (relatively few) may be distributed during the semester.

The following books are of particular interests to students enrolled in this class:

- Michael R. Lyu, editor, "Handbook of Software Reliability Engineering", McGraw-HIll, Inc.

  This book is available at the SMU Bookstore (214-768-2435). A CD-ROM containing software reliability tools and actual data is included in the book.

- J. D. Musa, A. Iannino and K. Okumoto, "Software Reliability: Measurement, Prediction, Application", McGraw-HIll, Inc.

  There are two eiditions, original (1987) and professional (1990). You may want to get the original edition for in-depth mathematics used in reliability models.

- N. E. Fenton and S. L. Pfleeger, "Software Metrics: A Rigorous and Practical Approach", PWS Publishing Company.

  Good treatment of measurement theory as applied to software metrics.

3. Workload and Grading

The course grade each student receives will reflect the weighted average of homeworks, a major project, and other assignments:

- *Project*: The project will consist of an application of a specific technique discussed in class in students' work and the report of related findings, or a term paper covering some in-depth study of a chosen topic. You may form teams to do the project. However, if you choose to write a term paper, it should be an individual effort. An oral presentation as well as a written report is required for each project.
- *Homeworks*: between 2 to 4 homeworks will be assigned throughout the semester.
- *Individual study presentation*: The in-class and TAGER students are required to give a presentation in class, lasting between 30 minutes and 3 hours (it's up to you) on a topic selected from our binder or from external sources (preferred) with my approval (maybe from your homework). The videotape students will be asked to prepare a presentation package with a detailed script.

The approximate weight assignment is as follows:

Project	50%
— 10%: proposal due before Spring Break	
— 15%: oral presentation before the last day of class	
— 25%: report due by the last day of class	
Homeworks	25%
Individual study presentation	25%

4. Course Contents and List of Papers

Part I. Software reliability engineering and recent development

We will study theory and practice of software reliability engineering (SRE) from various different perspectives. Topics to be covered include: 1) application environment and customer operational profiles, 2) testing strategy, process, and data collection, 3) software reliability growth models (SRGMs) and other analysis techniques, 4) applications, 5) recent development and research at SMU.

1. J. D. Musa and W. W. Everett. Software-reliability engineering: Technology for the 1990s. *IEEE Software*, pages 36-43, Nov. 1990.
2. A. L. Goel. Software reliability models: Assumptions, limitations, and applicability. *IEEE Trans. on Software Engineering*, 11(12):1411–1423, Dec. 1985.
3. N. F. Schneidewind and T. W. Keller. Application of reliability models to the space shuttle. *IEEE Software*, pages 28–33, July 1992.
4. J. Tian, P. Lu, and J. Palma. Test execution based reliability measurement and modeling for large commercial software. *IEEE Trans. on Software Engineering*, 21(5):405–414, May 1995.

5. J. Tian. Integrating time domain and input domain analyses of software reliability using tree-based models. *IEEE Trans. on Software Engineering*, 21(12):945–958, Dec. 1995.

6. M. H. Chen, M. R. Lyu, and W. E. Wong. An empirical study of the correlation between code coverage and reliability estimation. In *Proc. 3rd International Software Metrics Symp.*, pages 133–141, Berlin, Germany, 1996.

*. Additional material from the recommended textbooks for CSE 6340 will also be discussed.

Part II. Software measurement and analysis

Two groups of topics will be covered: 1) software measurement frameworks (Basili's GQM), 2) statistical analysis techniques (regression analysis, reliability growth modeling, principle component analysis, and tree-based modeling, etc.).

1. V. R. Basili and H. D. Rombach. The TAME project: Towards improvement-oriented software environments. *IEEE Trans. on Software Engineering*, 14(6):758–773, June 1988.

2. T. M. Khoshgoftaar, E. B. Allen, K. S. Kalaichelvan and N. Goel, Early quality prediction: A case sutdy in telecommunications. *IEEE Software*, pages 65–71, Jan. 1996.

3. W. N. Venables and B. D. Ripley. Tree-based methods. In *Modern Applied Statistics in S-Plus*, chapter 13, pages 329–348. Springer-Verlag, 1994.

4. A. A. Porter and R. W. Selby. Empirically guided software development using metric-based classification trees. *IEEE Software*, pages 46–54, Mar. 1990.

5. L. C. Briand, V. R. Basili, and C. J. Hetmanski. Developing interpretable models with optimal set reduction for identifying high-risk software components. *IEEE Trans. on Software Engineering*, 19(11):1028–1044, Nov. 1993.

Part III. Formal models for software metrics evaluation and selection.

This include several closely related topics: 1) "desirable" properties of metrics, 2) theoretical basis from measurement theory, 3) axiomatic models and classification scheme for metrics, 4) metrics evaluation and selection.

1. R. E. Prather. An axiomatic theory of software complexity measure. *IEE The Computer Journal*, 27(4):340–346, 1984.

2. E. J. Weyuker. Evaluating software complexity measures. *IEEE Trans. on Software Engineering*, 14(9):1357–1365, 1988.

3. J. C. Cherniavsky and C. H. Smith. On weyuker's axioms for software complexity measures. *IEEE Trans. on Software Engineering*, 17(6):636–638, June 1991.

4. J. Tian and M. V. Zelkowitz. A formal program complexity model and its application. *Journal of Systems and Software*, 17(3):253–266, Mar. 1992.

5. N. Fenton. Software measurement: A necessary scientific basis. *IEEE Trans. on Software Engineering*, 20(3):199–206, Mar. 1994.

6. J. Tian and M. V. Zelkowitz. Complexity measure evaluation and selection. *IEEE Trans. on Software Engineering*, 21(8):641–650, Aug. 1995.

7. L. C. Briand, S. Morasca, and V. R. Basili. Property-based software engineering measurement. *IEEE Trans. on Software Engineering*, 22(1):68–86, Jan. 1996.

5. General Sources for Software Engineering/Quality Literature

- Journals:
  - IEEE Transactions on Software Engineering (TSE).
  - ACM Transactions on Software Engineering and Methodology (TOSEM).
  - Journal of Systems and Software (JSS).
  - Annals of Software Engineering (ASE).
  - Software Quality Journal (SQJ)
  - Software Practice and Experience (SPE).
  - (IEE/BCS) The Software Engineering Journal (SEJ).
  - (IEE/BCS) The Computer Journal (CJ).
  - ACM Transaction on Programming Languages and Systems (TOPLAS).
  - IEEE Transactions on Reliability.
- Magazines:
  - IEEE Software.
  - IEEE Computer.
  - Communications of the ACM.
  - ACM SIGSOFT: Software Engineering Notes.
  - ACM SIGPLAN: SIGPLAN Notices.
  - IEEE Spectrum.
- Conference Proceedings:
  - International Conference on Software Engineering (IEEE/ACM: ICSE).
  - International Conference on Software Quality (ASQC: ICSQ).
  - International Symposium on Software Reliability Engineering (IEEE: ISSRE).
  - International Symposium on Software Testing and Analysis (ACM: ISSTA).
  - ACM-SIGSOFT Conference on the Foundational of Software Engineering (FSE).
  - International Conference on Computer Software and Applications (IEEE: COMPSAC)

6. Alphabetical listing of all the papers for CSE6340

[1] V. R. Basili and H. D. Rombach. The TAME project: Towards improvement-oriented software environments. *IEEE Trans. on Software Engineering*, 14(6):758–773, June 1988.

[2] L. C. Briand, V. R. Basili, and C. J. Hetmanski. Developing interpretable models with optimal set reduction for identifying high-risk software components. *IEEE Trans. on Software Engineering*, 19(11):1028–1044, Nov. 1993.

[3] L. C. Briand, S. Morasca, and V. R. Basili. Property-based software engineering measurement. *IEEE Trans. on Software Engineering*, 22(1):68–86, Jan. 1996.

[4] M. H. Chen, M. R. Lyu, and W. E. Wong. An empirical study of the correlation between code coverage and reliability estimation. In *Proc. 3rd International Software Metrics Symp.*, pages 133–141, Berlin, Germany, 1996.

[5] J. C. Cherniavsky and C. H. Smith. On weyuker's axioms for software complexity measures. *IEEE Trans. on Software Engineering*, 17(6):636–638, June 1991.

[6] N. Fenton. Software measurement: A necessary scientific basis. *IEEE Trans. on Software Engineering*, 20(3):199–206, Mar. 1994.

[7] A. L. Goel. Software reliability models: Assumptions, limitations, and applicability. *IEEE Trans. on Software Engineering*, 11(12):1411–1423, Dec. 1985.

[8] T. M. Khoshgoftaar, E. B. Allen, K. S. Kalaichelvan and N. Goel, Early quality prediction: A case sutdy in telecommunications. *IEEE Software*, pages 65–71, Jan. 1996.

[9] J. D. Musa and W. W. Everett. Software-reliability engineering: Technology for the 1990s. *IEEE Software*, pages 36-43, Nov. 1990.

[10] A. A. Porter and R. W. Selby. Empirically guided software development using metric-based classification trees. *IEEE Software*, pages 46–54, Mar. 1990.

[11] R. E. Prather. An axiomatic theory of software complexity measure. *IEE The Computer Journal*, 27(4):340–346, 1984.

[12] N. F. Schneidewind and T. W. Keller. Application of reliability models to the space shuttle. *IEEE Software*, pages 28–33, July 1992.

[13] J. Tian. Integrating time domain and input domain analyses of software reliability using tree-based models. *IEEE Trans. on Software Engineering*, 21(12):945–958, Dec. 1995.

[14] J. Tian, P. Lu, and J. Palma. Test execution based reliability measurement and modeling for large commercial software. *IEEE Trans. on Software Engineering*, 21(5):405–414, May 1995.

[15] J. Tian and M. V. Zelkowitz. A formal program complexity model and its application. *Journal of Systems and Software*, 17(3):253–266, Mar. 1992.

[16] J. Tian and M. V. Zelkowitz. Complexity measure evaluation and selection. *IEEE Trans. on Software Engineering*, 21(8):641–650, Aug. 1995.

[17] W. N. Venables and B. D. Ripley. Tree-based methods. In *Modern Applied Statistics in S-Plus*, chapter 13, pages 329–348. Springer-Verlag, 1994.

[18] E. J. Weyuker. Evaluating software complexity measures. *IEEE Trans. on Software Engineering*, 14(9):1357–1365, 1988.