Software Testing and Quality Assurance
First Semester, 2004

Instructor:
Taratip Suwannasart (ผศ.ดร. ธาราทิพย์ สุวรรณศาสตร์)

Office: Engineering Building 4, Room: 18-03, Phone: 02-218-6975
E-mail address: taratip.s@chula.ac.th
URL: http://www.cp.eng.chula.ac.th/~tsw

Grading:
• Midterm Exam - 30%  (August 6, 2004)
• Final Exam - 30%  (September 24, 2004)
• Homeworks and/or project – 40%


References:
Outline of Topics:

1. Course Overview
2. Software Testing and Software Development Life Cycle Models
3. Introduction
   - Historical views of software testing
   - Definitions
   - Principles of testing
   - Purpose, Goals, and Economics of Testing
4. Functional Testing
   - Boundary Value Testing
   - Equivalence Class Testing
   - Decision Table-Based Testing
5. Structural Testing
   - Path Testing
   - Data Flow Testing
6. Mutation Testing
7. Level of Testing
   - Unit Testing
   - Integration Testing
   - System Testing
   - Acceptance Testing
8. Object-Oriented Testing
9. Test Planning and Test Documentation
10. Managing the Testing Process
11. Test Organization
12. Testing Tools
13. Software Reviews, Inspections, or Walkthrough
14. Software Quality Assurance
1.2 Software Quality Management. Early after the introduction of the first computers and programming languages software became a critical for many organizations. The term software crisis was coined at the first NATO Software Engineering Conference in 1968 at Garmisch, Germany. Typically the crisis manifests in different ways including projects exceeding the estimated costs for development, the late delivery of software, and the low quality of the delivered software. Test cases and testing procedures should be formally designed and reviewed. Ad hoc testing is likely to miss potential defects in the software. Test cases should be used to demonstrate both the existence of software functionality and to attempt to uncover new defects. Testing, Quality Assurance, and Quantifiable Improvement. Jeff Tian. Department of Comp. Software Testing Testing Across the Entire Software Development Life Cycle. Mastering Software Quality Assurance is a comprehensive reference on quality assurance as it pertains to software develop. Software Testing: An ISTQB-BCS Certified Tester Foundation Guide 3rd ed. 278 Pages. 3 Quality Assurance. 3.1 Classification: QA as Dealing with Defects 3.2 Defect Prevention. 3.2.1 Education and training 3.2.2 Formal method 3.2.3 Other defect prevention techniques 3.3 Defect Reduction 3.3.1 Inspection: Direct fault detection and removal 3.3.2 Testing: Failure observation and fault removal 3.3.3 Other techniques and risk identification 3.4 Defect Containment 3.4.1 Software fault tolerance 3.4.2 Safety assurance. and failure containment 3.5 Concluding Remarks Problems. 4 Quality Assurance in Context. This book evolved from class notes for the one-semester course Software Testing and Quality Assurance that I have taught many times at Southern Methodist University since 1995. Software quality assurance (SQA) is a means and practice of monitoring the software engineering processes and methods used in a project to ensure proper quality of the software. It may include ensuring conformance to standards or models, such as ISO/IEC 9126 (now superseded by ISO 25010), SPICE or CMMI. It includes standards and procedures that managers, administrators or even developers may use to review and audit software products and activities to verify that the software meets quality. A man checks under a tank. A tank is pulled slowly.