17-654 Analysis of Software Artifacts
Instructor: Dr. Eduardo Miranda
Spring 2016

On August 8th, 2012 James Kwak, an associate professor at the University of Connecticut School of Law wrote an article on the magazine The Atlantic entitled: "*Software Runs the World: How Scared Should We Be That So Much of It Is So Bad?"*[1]

The Analysis of Software Artifacts course attempts to answer this question and to provide techniques to develop confidence on the quality of the software being produced or acquired.

The course adopts the view that software quality is not only the absence of defects but it encompasses all the characteristics that bear on the its ability to satisfy stated and implied needs. Software quality is then defined from different perspectives: product quality, quality in use and process quality through the use of specific quality models. The course systematically explores different quality attributes and the techniques most appropriate to verify them. Specific topics include software testing, static analysis and model checking, inspections, technical debt, cost of software quality, planning for quality, quantitative models and defect classifications.

The course balances traditional lectures with small projects in which students apply the ideas they are learning to real artifacts. The final project consists on the preparation of a quality plan for the Studio project.

Prerequisites:

Discrete math and statistics knowledge at the undergraduate is strongly recommended. Programming experience and familiarity with the Java language are mandatory for the completion of the assignments.

Lecture Topic	**Mandatory readings**	**Optional readings (To learn more)**
Introduction, Views on quality	Garvin, What Does "Product Quality" Really Mean? Kruger, Main Schools of TQM	Walden, Kano's Methods for Understanding Customer-defined Quality
Quality models	Naik, Software Quality	Polillo, Quality Models for Web Sites

[1] http://www.theatlantic.com/business/archive/2012/08/software-runs-the-world-how-scared-should-we-be-that-so-much-of-it-is-so-bad/260846/

Technical Debt: Measurement & quality metrics (1)	McConnell, Managing Technical Debt Letouzey, Managing Technical Debt with the SQALE Method Chin, Economics of Technical Debt	Letouzey, The SQALE Method Definition Document Fairley, What Should Be Measured
Measurement & quality metrics (2)	Hubbard, The Illusion of Intangibles SEI, Foundations of Measurement	Buse, Software Analytics Rosemberg, Statistics and Measurement
Verification overview	Rybers, Essential Testing: Chapters 5 & 7	
Testing - Measuring test adequacy	Buechner, Is 100% Code Coverage Enough? Marick, How To Misuse Coverage Smith, Should Software Testers Use Mutation Analysis to Augment a Test Set?	Wei, Is Coverage a Good Measure of Testing Effectiveness?
Testing - Selected black box techniques (1)	Rybers, Essential Testing: Chapters 10 & 13	Marick, Experience with the Cost of Different Coverage Goals
Testing - Selected black box techniques (2)	Rybers, Essential Testing: Chapters 14 & 16	
Testing - Modeling the input domain	Buechner, Test Case Design Using the Classification Tree Method	Perry, Object-Oriented programs and Testing
Combinatorial testing	Rybers, Essential Testing: Chapter 6	Schieferdecker, Model-Based Testing
Testing - Basis path testing & data flow testing	McCabe, Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric Naik, Data Flow Testing	

Testing - Random & mutations	Naik, Random Testing Hamlet, Random Testing Warnok, Look out! It’s the fuzz!	Hamlet, When Only Random Testing Will Do Pacheco, Finding Errors in .NET with Feedback-Directed Random Testing
Inspections	Wiegers, Peer Reviews in Software DoD, Formal Inspections Cohen, Questions for Review Process Ford, Process for Performing Security Reviews	Aurum, State-of-the-art: software inspections after 25 years Malotaux, Inspection Manual: Procedures, rules, checklists and other texts for use in Inspections
Static Analysis & Model Checking (1)	Chess, Secure Programming with Static Analysis: Chapters 2 & 3	Chess, Secure Programming with Static Analysis: Chapter 4
Static Analysis & Model Checking (2)	Mordechai (Moti) Ben-Ari, Primer on Model Checking NASA, Program Model Checking: A Practitioner’s Guide:Chapters 1, 2 & 7	NASA, Program Model Checking: A Practitioner’s Guide: Chapter 4
Performance analysis & verification (1)	Jewell, Performance Engineering and Management Method —A Holistic Approach to Performance Engineering Hillston, Operational Laws	Little, Little's Law David, Timed PetriNets Dutta, Systems Modeling for IP-based Handoff Using Timed Petri Nets Lazowska, Quantitative System Performance
Performance analysis & verification (2)	Shallahamer, Practical Queuing Theory Everett, Performance Testing	Gunther, Performance Measurements and Tools Ball, Using Paths to Measure, Explain, and Enhance Program Behavior Subraya, Performance Counters In OS
Quality planning (1)	Galin, Cost of Software Quality Wagner, Software Quality Economics for Combining Defect-Detection Techniques Shenvi, Defect Prevention with Orthogonal Defect Classification	Freimut, Developing and Using Defect Classification Schemes Huber, Comparison of IBM’s ODC to Hewlett Packard’s Defect Origins, Types, and Modes El-Emam, The ROI from Software Quality An Executive Briefing

Quality planning (2)		
Security analysis & verification: Software vulnerabilities and exploitation	Mell, Complete Guide to the Common Vulnerability Scoring System; Nist, Technical Guide to Information Security Testing and Assessment	Avizienis, Basic Concepts and Taxonomy of Dependable and Secure Computing (To learn more); OWASP, Testing Guide (To learn more)
Organizational quality & Process improvement (1)	SEI, CMMI for Development, Version 1.3: Part 1, Generic Goals and Generic Practices, Project Planning (PP) Process Area & Project Monitoring and Control (PMC) Process Area	
Organizational quality & Process improvement (2)	Mutafelija, Appraisal and Registrations Weller, Practical Applications of Statistical Process Control Weber, Theory of Constraints Raisinghani, Six Sigma: concepts, tools, and applications	SEI, Practical Software Measurement: Measuring for Process Management and Improvement

Policies for 17654 – Analysis of Software Artifacts

A. Grading general

Grading is accomplished through the instruments below. The weights assigned to each instrument are designed to achieve a balance between individual and group activities as well as hands-on, reading and class discussions.

1) Midterm exam: 20%
2) Final exam: 20%
3) Quality plan: 10%
4) Group projects excluding quality plan: 20%
5) Individual Assignments: 20%
6) Participation: 10%

All items will be graded in a scale from 0 to 100 points.

Undergraduates taking this course are exempt from working on the quality plan project and in their case the weight for the individual assignments will be 30% of the total

B. Exams

To the effect of ensuring consistency among the graders, the questions to be included in the exams should have a clear right/wrong answer that can be evaluated through a rubric.

Partial points will only be only given for incorrect computations or single missing items.

Responses that are contradictory, non-falsifiable or grossly incomplete will get 0 points.

C. Quality plan

The quality plan is a semester long group project. The total work is divided into three milestones.

Milestones one and two are worth 15 points each and are graded in a pass/no pass fashion. These milestones are selected to reinforce the concepts learned in class and to avoid the rush to work in the plan at the end of the semester.

Milestone three has a maximum of 70 points that are assigned on the basis of quality of content and presentation.

D. Group projects

Group projects are designed to encourage reflection and exploration of the concepts explained in class through hands-on activities. Failure to follow the submission guidelines will result in a 10 points penalty.

Students will provide feedback about teammates. Students mentioned by three or more colleagues as not helping will lose 10 points from the assignment and will see their participation grade affected

The final grade for the group projects is the average of each of the projects. The grading of each project will reflect the quality of the content as well of the presentation in class.

E. Individual assignments

Individual assignments are designed to keep the student engaged and verify the level of understanding as the course progress. There are seven individual assignments. Failure to follow the submission guidelines will result in a one point penalty.

As a general rule when a question or task in the assignment asks you to describe something in your "own words" copying and paste is not a good option. We want you to paraphrase ours or somebody else material to demonstrate a certain level of understanding.

The final grade for the individual assignments is the average of the best 6 assignments.

F. Participation

The participation component of the grade is designed to encourage attendance and class discussion. Participation points range from 0 to 100. The grade is faculty discretionary, based on the consensus of the instructors and TA

- Everybody gets 50 points to start with
- You can grow it or you can deplete them
- A non-exclusive list of things that negatively affects your participation grade:
    - No attending lectures
    - Being late to class
    - Academic integrity incidents
    - Not working on team assignments
- Things that contribute towards participation:
    - Critical thinking
    - Asking questions
    - Helping others
    - Mindfully asking questions

G. Late submittals

All assignments must be submitted by 5:00 PM on the date they are due. Late submittals are not allowed. The policy of discarding the worst assignment compensates for this. The student is welcomed to the recitations where it will get feedback if so desires.

H. Academic integrity violation

There is a zero tolerance policy for academic misconduct. Any incident deemed as such by the faculty will result in the following:

1) The forfeiture of the grade corresponding to the incident in what the violation took place
2) A zero in participation
3) Reporting of the violation to the proper authorities

I. Classroom etiquette

Please keep in mind that your conduct has an impact on others. You are not required to attend classes; although not doing so will impact your participation grade. If you do attend, however, certain behaviors are unacceptable and discourteous to the instructor and your classmates. Among them sleeping, texting, reading, and talking to classmates during lecture.

The class will begin promptly at the scheduled time and you are expected to be in your seat at that time. Please seat always in the same place seat this is how we track your participation. Changing seats might result in your contribution being unnoticed.

Use of headphones, tablets, smart phones, laptop computers, or any other electronic equipment is NOT allowed during class without instructor authorization.

Failure on your part to observe these guidelines will result in us asking you to leave the class for the remainder of the period.