

Spring 2020

CS 684-102: Software Testing and Quality Assurance

Samir Padalkar

Follow this and additional works at: <https://digitalcommons.njit.edu/cs-syllabi>

Recommended Citation

Padalkar, Samir, "CS 684-102: Software Testing and Quality Assurance" (2020). *Computer Science Syllabi*. 83.

<https://digitalcommons.njit.edu/cs-syllabi/83>

This Syllabus is brought to you for free and open access by the NJIT Syllabi at Digital Commons @ NJIT. It has been accepted for inclusion in Computer Science Syllabi by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.



Syllabus: CS 684, Spring 2020

Software Quality and Testing

Course Description

CS 684 - Software Quality and Testing (3 credits)
This course discusses the general topic of defining software quality attributes and deploying techniques to ensure that these quality attributes are met. Focus is placed on functional quality attributes, such as correctness, reliability, safety, security, etc. A variety of techniques are considered, by virtue of the law of diminishing returns: requirements specifications, static code analysis, software testing, program verification, and software reliability modeling. Advanced topics such as Agile Development & Testing, Testing Large-Scale Systems such as Cloud-Based, Big-Data, IoT, AI; shall also be covered.

Personnel

Samir Padalkar, Office:	
Fax: 973-596-5777	Homepage:
Phone: 973-596-3366	Email: padalkar@njit.edu
TA: TBD	

Instructor Availability and Response Time

Please include cs684 in the Subject: line of any email you send; it will make it easier to manage my emails. I will do the same in emails I send to you. I will check email regularly. If I do not respond to your email within 48hrs (I usually respond within 24 hours or

sooner) during the weekdays, please send the email again. If you need to see me, we can meet after class or by appointment. I will make every effort to post your grades a week after they're due.

Learning Outcomes

- Understand the overall process of Software Testing & Quality Assurance
- Acquire direct, hands-on experience specifying requirements, developing code, & testing code; for an advanced project (DB, Servers, UI, Computations & Logic, Real-Time, A.I., Advanced Data Science ...)
- Understand how to develop & implement Unit Tests, Integration Tests & System Tests
- Understand how to develop & implement Test Generation & Test Oracles
- Acquire hands-on experience with Software Testing Tools; especially in Code Coverage, Performance Testing, & General Testing
- Understand & appreciate complexities in Software Testing and how to handle them

All course materials, including lecture notes, papers, etc, will be posted on Moodle. Announcements and notices will also be posted. **WHEN IN DOUBT, CHECK Moodle.** Moodle will be used for online Q&A and for any paper or presentation submissions

Simple rules

Turn off the toys (phones especially).
I expect you to behave professionally.
Speak to me about any issues you have.

Ethical Conduct (This should go without saying, people. Seriously.)

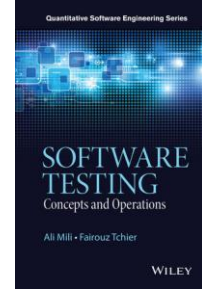
Cheating during in-class tests or take-home examinations or homework is, of course, illegal and immoral.

The essential quality of the NJIT **University Code on Academic Integrity** is that each student shall demonstrate honesty and integrity in the completion of all assignments and in the participation of the learning process. Adherence to the University Code on Academic Integrity promotes the level of integrity required within the university and professional communities and assures students that their work is being judged fairly with the work of others. See <http://www.njit.edu/academics/pdf/academic-integrity-code.pdf>

Readings

- 1) *Software Testing: Concepts and Operations*. Ali Mili, Fairouz Tchier. ISBN: 978-1-118-66287-8. 368 pages. April 2015.

<http://as.wiley.com/WileyCDA/WileyTitle/productCd-1118662873.html>.



2) *How Google Tests Software* : James Whittaker, Jason Arbon, Jeff Carrolo

<https://www.amazon.com/Google-Tests-Software-James-Whittaker/dp/0321803027>

Assignments

Students are required to form groups of four members each, and each group will deploy the techniques discussed in class for a full-length software lifecycle, from specification generation down to programming, testing and reliability estimation. It is expected that students combine assignments & team project into a more complex and real-life exercise. A combined more-complex team project, with weekly submissions will be 70 % of the final grade.

Grading:

Students are graded on the basis of the following criteria.

Team Projects	70
Final Exam	30
Total	100

Important Dates:

27-Jan	First Day of Class
3-Feb	Project Team and Topic Due
24-Feb	Project Requirements, Architecture, & Specs Due
24-Feb	Software Testing Tool Choice for Demo Due
9-Mar	Software Testing Tool Demo
23-Mar	Project Final Working Code Due
30-Mar	Project Working Code Demo
20-Apr	Project Testing Code Due & Testing Demo
4-May	Final Project Due
4-May	Final Peer & Self Project Evaluation Due
4-May	Final Project Demo
11-May	Final Exam (I Think)

Course Schedule

COURSE SCHEDULE		
Week	TOPIC	IN CLASS
JAN 27	Introduction to Software Testing	Project Explanation
FEB 3	Software Specifications	Past Final Paper Section
FEB 10	Agile Development & Testing	Past Final Paper Section
FEB 17	SW Fault Types. Unit / Integration / System Tests	Past Final Paper Section
FEB 24	GUI / DB / Server/ Data Science Testing	Past Final Paper Section
MAR 2	Software Testing Taxonomy	GUI/DB/Srvr/DataSci Testing
MAR 9	Demo : Software Testing Tools	Demo Continues
MAR 16	Spring Break	
MAR 23	Test Data Generation, Functional Criteria	GUI/DB/Srvr/DataSci Testing
MAR 30	Demo : Working Code-I	Demo Continues
APR 6	Structural Criteria, Oracle Design	Help with Project
APR 13	Test Driver Design, Test Outcome Analysis	Help with Project
APR 20	Demo : Working Code-II, Testing-I	Demo Continues
APR 27	Cloud, Big Data, AI, IoT Testing	Help with Project
MAY 4	Demo : Testing & Working Code - FINAL	Demo Continues
MAY 11	Final Exam (probably)	Material similar to Past Exam

Project Schedule

PROJECT SCHEDULE		
Week	Topic	Description
JAN 27	Introduction	
FEB 3	Project Team & Topic Due	Team Consisting of 4 students. Project Topic
FEB 10	User Requirements Due	High Level Version
FEB 17	Architecture & Class Design Due	Class & methods per student. Each student has separate class
FEB 24	Specifications Due. Testing Tool choice for Demo due	Detailed specs for each class & method. Choice of Testing Tool for presentation & demo due.
MAR 2	Working Code Version 1 Due	Interim Code, does not have to compile/run
MAR 9	Working Code Version 2 Due	Header files complete. Some code complete
MAR 16	Working Code Version 3 Due	Has to compile, some functionality should work
MAR 23	Final Working Code Due	Should be working, at least an alpha version
MAR 30	Test Version 1 Due	Overall Strategy & Design
APR 6	Test Version 2 Due	Test Oracle 1 st Version & 25 % Tests Complete
APR 13	Test Version 3 Due	Test Oracle 2 nd Version & 50 % Tests Complete
APR 20	Test Version 4 Due	Test Driver 1 st Version & 75 % Tests Complete
APR 27	Final Project Due (1 st Version)	Test Data Set, Test Oracle, Test Driver complete
MAY 4	Final Project Due (Final Version)	ALL IS DUE (FINAL)

Academic Integrity

Please be advised that academic integrity and honesty are of paramount importance to me. I will uphold NJIT's honor code, and will have the duty to report any violation of this code to the immediate attention of the Dean of Students.

Attendance

A great premium will be placed on attendance and active participation in class discussions. A record will be kept of class attendance; and may be used at the end of the semester to draw lines between letter grades. Anyone who signs the attendance sheet commits to attend the whole class.

Grading Policy

Grading Policy shall follow that outlined in <http://www.njit.edu/registrar/policies/grading.php>

Graduate Grading Legend

Grades	GPA	Significance
A	4.0	Excellent
B+	3.5	Good
B	3.0	Acceptable
C+	2.5	Marginal Performance
C	2.0	Minimum Performance
F	0.0	Failure

Grading is on a curve, with the following scale as a guide

A = 90 to 100

B+ = 80 to 89

B = 70 to 79

C+ = 60 to 69

C = 50 to 59

As per NJIT CS Dept. guidelines, the following is supposed to be the distribution of grades ---

- 1) A = Top 25 % (or less) of class
- 2) B+ = Next 25 % (or less) of class
- 3) B = Next 25 % (or less) of class
- 4) C+/C/F = Next 25 % (or remaining) of class

MAKE-UP EXAM POLICY

Make Up Exam policy is based upon generally accepted policies of the NJIT CS Dept., and individual circumstances.

PROJECT GRADING

Version 1.1

A) Project Marks / Total = 70/100

B) Breakdown ---

1) Requirements – 5/70

2) Architecture + Working Code – 20/70

3) Testing Tool Presentation & Demo – 5/70

4) Testing Code – 40/70

B.1) Requirements ---

a) All Teams shall get at least 4/5 if they do an honest submission.

b) Teams may repeatedly improve this submission until end of term.

c) 5/5 is reserved for the team that does a professional job.

B.2) Architecture + Working Code

a) OnTime = 4/20; Complexity = 5/20; Works = 11/20

b) OnTime implies 4 submissions have to be On Time.

c) Complexity in Architecture & Implementation is further defined later.

d) Works implies How Many “Complex” & “Simple” Use-Cases are “Alpha Level Working”.

e) “Works” & “Complexity” will be judged first on Mar 20; and finally on May 1.

B.3) Testing Tool Demo ----

a) All Teams should get 5/5, if they do an honest presentation (i.e. match expectations).

B.4) Testing Code ---

a) OnTime = 6/40; Complex = 9/40; Works = 25/40

b) OnTime implies 6 submissions have to be On Time.

c) Complexity in Tests is further define later.

d) Works implies at least 20 Tests.

C) Project will be treated like a real-life AGILE project at a Software Company

D) There are weekly deliverables, as specified in the schedule. Penalties will be assessed for missing and/or late deliverables.

E) Complexity of Project will be given supreme importance. The more complex and ambitious and NEW the Project, THE BETTER. For example, Projects in the following fields are encouraged –

- 1) Big-Data
- 2) Artificial Intelligence
- 3) Advanced Data-Science
- 4) Advanced Real-Time
- 5) IoT (Internet of Things)

- 6) Security
 - 7) Advanced Bio-Informatics
 - 8) Other similar areas
 - 9) A Combination of any of the above
- F) It is OK to use public-domain source code in your project, so long as it is properly credited & specified.
Apache Products (and other Open Source) such as Hadoop, SPARK, Kafka, Lucene, Tomcat etc. are strongly urged.
- G) It is OK to use code written for other projects in your project, so long as it is properly credited & specified.
- H) Project Code can be written in any of the following languages --- C, C++, JAVA, SCALA, R, PYTHON, LISP, C#, FORTRAN, PASCAL, ADA etc.
- I) Project may run on Windows and/or Linux.
- J) Project may use Relational and/or NoSQL and/or Big-Data DB's.
- K) Project should use at least one commercial and/or Open Source Testing Tool.
- L) General Expectation about Project Code (just Project Code & NOT Testing Code) --- Each student is expected to write at least 750 to

2,000 lines of code. This is your own code, and NOT public domain and earlier project code.

- M) Final Working Code is expected to be “ALPHA”. A few Use Cases should work, and be demo-able, and should not crash. It is OK to have BUGS in your Final Working Code; ideally these should be discovered in Testing Code.
- N) Although Testing Code is due after Final Working Code; Good AGILE Testing implies that UNIT Tests should be supplied with working code.
- O) A Presentation & Demo of 1 Testing Tool is scheduled for Mon March 6 2017. Each Project should present the following about the Testing Tool they have chosen ---
 - 1)Overview
 - 2)What does it do, and what is it Good For
 - 3)Use Case (Example)
 - 4)Advantages
 - 5)Disadvantages

The choice of Testing Tool is due with Software Specifications, on Mon Feb 20 2017.

- P)A separate “Expectations” document describes expectations about Working Code & Testing Code.

- Q) Completeness, Innovation, Uniqueness, Cutting-Edge, Complexity, Works etc. are major characteristics on which the Final Working Code & the Final Testing Code will be judged.
- R) Each individual's contribution to the project shall be assessed separately. Individuals not pulling their weight in the project shall be penalized. A Confidential Survey shall be given to ALL students, and will ask each one to grade the other students in their common project.

EXPECTATIONS

- A) REQUIREMENTS**
- Each student is responsible for at least 1-2 major classes. Each class should contain at least 3-6 major methods.
 - Hence each student is responsible for at least 6 – 10 major methods.
 - A major method can be created from several simple methods. I am NOT looking for each major method to be just 1 method; but as good software engineering dictates; a major method should be composed of several smaller methods.
 - Each major method should be reproducible from a requirement. In other words; the total number

of functional requirements per student should roughly match the number of major methods.

- Each student should specify 2-5 additional other requirements such as performance, latency, security
- The definition of a “major method” is flexible.

Guidelines include ---

1. Internally should use a major Data Structure such as vector, array, list, set, hash-table, B-Tree etc.
2. Implements relatively complex logic/numerical algorithm.
3. Implements relatively complex DB stored-procedure
4. Implements relatively complex mapping of one data structure to another data structure
5. Implements relatively complex GUI widget get/set values.
6. Implements/Parses a major section of a web-page.
7. Implements relatively complex DB stored-procedure

B) TESTING

- Each student is responsible for at least 20 test cases.
- A sample division amongst these 20 test cases is
 1. 10 unit tests

2.5 integration tests

3.5 system tests

- At least 20 test cases have to be implemented in code.
- Test Cases have to cover expected inputs, and bad inputs.
- Test Cases have to test functionality (GUI, DB, Logic/Numerical ...) and other (performance, latency...)
- Test Cases have to test worst-case in terms of number (million, billion, trillion ...)
- Test Cases have to be relatively unique.

GRADING FOR TESTING CODE

Grading for Testing Code is as follows (total of 40 points)

- a) OnTime - 6/40
- b) Use of 3rd Party Testing Tool = 5/40
- c) Working Test Cases = 20/40
- d) Complexity = 9/40

3rd PARTY TESTING TOOL

Team usage of 3rd party tools for Unit Tests, Code Coverage, & Performance/Load Tests (appx. 1.5 each)

WORKING TEST CASES

At least 20 Working Test Cases, comprising of 10 Unit Tests, 5 Integration Tests, & 5 System Tests; have to be submitted.

- 1) 1 point for each WORKING UNIQUE test case.
- 2) WORKING implies working test code. A few complex tests cases could be submitted as just outline/description.
- 3) UNIQUE implies no one else in team has the same test case.

COMPLEXITY

Complexity is based on the following factors

- 1) Relative to ALL students
- 2) Complexity of Working Code
- 3) Functional Requirements Coverage (Good inputs, Bad inputs, average Use-Case, Large-Scale Use Case)
- 4) Performance & Load Testing
- 5) Other Requirements Testing (Availability, Reliability, Security, Usability, Multiple Environments ...)
- 6) Test Case Generation Strategy (Manual, Random, Statistical, Heuristic, Automatic ...)
- 7) Writing Own Test Data/Situation Simulators/DB-Data-Loaders
- 8) Writing Own Stubs/Mocks

- 9) Test Cases to detect complex bugs/issues such as Concurrency Issues(Deadlocks, Race Conditions ...), Math issues (Divide by 0, Boundary Conditions, Convergence, ...), GUI issues(Multiple Pages, Multiple States, Multiple & Complex Widgets, Patterns ...), Server issues(Scalability, Communications, API, Multiple States, Complex Algorithms, Memory usage, Data Structures, Patterns, Security ...); DB issues (Scalability, Partitioning/Sharding, Parallelism, Patterns, Complex Queries...).
- 10) Testing Oracle & Test Driver
- 11) Unique/Novel Contributions towards advancing SW Testing (Wrote own tools/plugins, Automatic Test Case Generation ...)

THIS DOCUMENT CONTAINS:

- 1) New Suggestions for Projects; Post CV Online CS684 Spring 2020
- 2) New Grading Criteria for Projects; Post CV Online CS684 Spring 2020
- 3) Final Exam CS684 Post CV Online Spring 2020

NEW SUGGESTIONS PROJECTS POST CV ONLINE CS684 SPRING 2020

Here are some suggestions that teams of 1 or 2 students can implement. These suggestions are also valid for larger teams, ie 3 or 4 students.

GUI:

- i) Create a GUI Window / Page with several different GUI widgets (similar to the Final Exam GUI question). Ensure that “user-entry” widgets are enabled only when needed.
- ii) Ensure that “multiple states” concept is implemented, either via one window / page or several window / pages. For example; different states in an airline reservation system could be: a) User Travel Choice entry (From Airport, To Airport, Dates etc) b) Schedule & Airline Choice & selection c) Payment & finishing touches.
- iii) Implement Navigation between Multiple Windows / Pages
- iv) Like Google, Break Results into Multiple Pages and provide an index at bottom (or elsewhere)
- v) Provide Visual Real-Time Data updates; stock prices changing, or number of logged users changing, or any other similar. Implement this feature either by periodically pulling data (polling), or responding to data asynchronously “pushed” out of a server
- vi) Implement Data Visualization windows / pages: graphs, charts etc.

SERVER:

- i) Implement Advanced Math / Data Science server, consisting of math / DS models. These models can be part of a comprehensive Business Logic server.
- ii) Implement In-Memory DB to speed up processing. Either implement own or use open-source MemCache or Redis.
- iii) Implement several business rules / constraints / algorithms

- iv) Implement Transaction Processing (Specify, get-choices, select, pay) (Financial Markets view real-time data, create & send bid/ask order, trade to update position & risk ..)
- v) Implement Bio-Informatics algorithms, Gene Editing, DNA/RNA/Amino-Acid analytics ...
- vi) Implement Real-Time Data Processing: Get Real-Time Data from other servers / external sources, Send Real-Time Data to other servers / DB / GUI. Ensure that concurrency principles are met, such as: no deadlocks, no race conditions, proper synchronization.
- vii) Implement a web-server; ie a server that allows external clients (browser or app based or TCP/IP-HTML native) to connect via TCP/IP Internet. Ensure that it processes requests from the connected GUI's, and send them proper responses.
- viii) Ensure that server interacts (Inserts, Updates, Deletes, Queries Data) with a proper DB, either SQL or NoSQL or Big-Data.
- ix) Implement Distributed / Parallel Processing: Several different types of Servers (Web, Biz-Logic, Real-Time-Data Input/Output, DB-Interface, Math-DS ...)
- x) Implement Load Balancing between several servers of same type; for example web-servers.
- xi) Implement Map-Reduce

DB:

- i) Ensure that large quantities of data are injected into DB. All code required to do this injection is part of DB Code (servers that access external data, real-time data servers, scripts that download & insert data etc ...)
- ii) Ensure that stored procedures are written for SQL DB's. These SP's should INSERT / UPDATE / DELETE / QUERY DATA.
- iii) Ensure that proper schema is created
- iv) Ensure that proper indices are created
- v) Implement backup

NEW GRADING CRITERIA PROJECTS POST CV ONLINE CS684 SPRING 2020

- A) PROJECT still stays at 70 / 100
- B) FINAL EXAM still stays at 30 / 100

PROJECT Grading Updates:

- 1) Requirement Docs stays at 5 / 70. Already graded. All students have received 4 / 5.
- 2) Testing Tool Demo stays at 5 / 70. Already graded. All students have received 5 / 5
- 3) OnTime Submission stays at 10 / 70. All students have received 4 / 4 for 4 assignments so far. This was the concession to the distress caused by the CV situation. The 6 future assignments shall be strictly checked for OnTime Submission. Hence, please submit on time.

- 4) Project Working Code stays at 11 / 70. Teams that have remained intact even after today, have to demo 8 Use-Cases (team of 4) (reduction from 10), or 6 Use-Cases (team of 3), or 3-4 Use-Cases (team of 2 or 1).
- 5) Project Working Code Complexity stays at 5 / 70. Measures of Complexity for intact teams, whether 3 or 4 students; & Measures of complexity for newly formed teams are similar:
 - i) Teams of 3 or 4 are expected to stick to the 3 tier architecture; ie GUI + SERVER + DB
 - ii) Teams of 1 or 2 can create just one program that has some user input & reads data from files. Obviously, if a team creates either a proper 2 TIER (GUI + SERVER, OR SERVER + DB) or a 3 TIER; then that is considered to be a major plus.
 - iii) Requirement of Amount of code written by each student stays roughly the same. I realize that the smaller teams (1 or 2) may have to write more lines of code; and they shall be given some more time.
 - iv) Other complexity measure such as using advanced data structures, advanced algorithms etc still remain the same.
- 6) Testing Tool Usage in Project remains at 5 / 70. Teams of 3 or 4 have to demo at least 3 types of testing tool usage: a) Unit Testing b) Code Coverage Testing & c) Performance Testing.
Teams of 1 or 2 students shall demo at least 2 of above 3 (a,b,c). If you demo the required, you shall get 5 / 5.
- 7) 20 Unique Test Cases gets you 20 /70. A unique Test Case is
 - A) A Test Case that tests a different part of a project OR
 - B) A Test Case that tests the same part of a project; but with different Data Inputs. For example testing “Login” with 20 different userid / passwords shall be considered as 20 unique test cases.
- 8) And finally, TESTING COMPLEXITY remains at 9 /70. Measures for Testing Complexity remain the same as before.
As repeatedly mentioned in class, 20 “Login” Test Cases shall get 20 / 20 for Test Cases; but shall get you 1 / 9 on complexity. This strategy can & should be used by students who get into trouble for whatever reason.

FINAL EXAM CS684 POST CV ONLINE SPRING 2020

- 1) Final Exam for CS684 Spring 2020 is currently scheduled MON May 11 6 pm to 8:30 pm
- 2) NJIT & the CS Dept has advised either Online Exam or an “Authentic Assessments”.
- 3) My reasoning for an Online Final Exam on MON May 11 6 pm to 8:30 pm:
 - a) My Exam are always Open-Book & Open-Internet, hence the Online Final Exam is a Take-At-Home Exam, Open-Book & Open-Internet, subject to Honor Code --- “No Help From Any Other Person”.
 - b) 70% of grade is allocated to the project, and it matches several goals of “Authentic Assessment”. From the NJIT site:
<https://www.njit.edu/ite/authentic-assessment>

“Authentic assessment evaluates whether the student can successfully transfer the knowledge and skills gained in the classroom to various contexts, scenarios, and situations. Authentic Assessment is grounded in theoretical best practices for teaching and learning and serves as an effective measure for course learning outcomes. In many ways, it can be considered the difference between measuring what students know vs. how they can apply that knowledge. These types of assignments will vary by discipline **but typically require students to complete a project.** For example, you may ask students to apply an engineering problem to a real world example, develop a web application, design a model, critically review case studies, or create multimedia presentations. “

- 4) Hence, unless NJIT & the CS Dept. issue any changes; the Final Exam for CS684 Spring 2020 is to be held as on Online Exam; on Mon May 11 6 pm to 8:30 pm. I shall upload the Final Exam Paper (as a Word Doc & as a PDF doc) onto Canvas at 5:55 pm, and expect students to download it, and answer it in 2.5 - 3 hours; and upload the answers back on canvas by 9 pm.

Students can a) Type answers into the Word Doc & upload it back. B) Print either the PDF or Word Doc, write answers with bold black / blue pen, scan and convert to a PDF doc & upload.

I shall strictly ensure a cutoff at 9 pm, students submitting later than 9 pm shall have points deducted based upon how late they submit. I shall be available to answer questions via email / canvas.