University of South Florida
Muma College of Business Administration
School of Information Systems and Management
ISM 4432 – Software Testing

Instructor:	Alan R. Hevner	*E-Mail:*	ahevner@usf.edu
TA/GA:	Sanjana Thinderu	*E-Mail:*	sanjanthinderu@usf.edu
Term:	Spring Semester 2023	*Dates:*	2/6/2023-4/28/2023
Delivery Method:	Classroom (CL)	*Location:*	BSN 120
Course Prerequisites/ Minimum Technical Skills & Requirements:	Course Prerequisite: ISM 6124 *In order to take courses online at USF, you will need to be able to demonstrate proficiency at basic computer skills, maintain reliable internet access, and meet the computer system requirements listed here:* http://www.usf.edu/innovative-education/resources/student-services/technical-requirements.aspx		
Office Hours:	Instructor – Monday 5-6pm in office (CIS 2076) and on MS Teams. Available also by appointment TA – By appointment		

I. Instructor Contact Information and Communication

The preferred method of instructor contact is via ahevner@usf.edu . You may address the instructor as Prof. Hevner.

II. First Week Attendance Policy

First week attendance will be taken during the first class period on February 6. If you are not marked present in first week attendance you will be dropped from the class.

III. University Course Description

The activities of software testing are among the most intellectually challenging in the field of software development. Skilled software testers are in great demand throughout the IT industry as we strive to produce high quality, reliable software system products and services. The processes and techniques of software testing attempt to verify the quality of software systems before they are released into the field. It is well known that one cannot test quality into software. Software quality is predicated on effective development and verification for requirements, specification, design, and implementation. Testing must be an integral component of all development processes to ensure an adequate level of quality. Software security issues are highlighted in all topics of the course.

IV. Course Purpose

Students will learn the fundamental concepts of quality assurance and software testing. The course will survey and analyze the best practices in industrial testing groups. New research ideas for improving testing will be explored. Students will gain practical experience with both functional (black box) and structural (clear box) testing methods via assignments. Automated testing tools will be an important part of the educational experience. The goal is for all students to achieve an in-depth understanding of software testing research and practice.

V. Course Structure

This course is structured as weekly modules in Canvas. The content of each module is specified in the course Syllabus and corresponding Canvas Modules. Assignments and grading criteria are provided in the Canvas assignments. Student discussion boards provide a weekly assessment of student understanding and progress.

VI. Course Objectives

Upon completion of this course, students will:

- Know the objectives of software testing to produce high quality software systems
- Understand the basic concepts and definitions used in the field of software testing
- Understand how quality assurance activities are applied throughout the software development life cycle

VII. Student Learning Outcomes

By the end of this course, you will be able to:

- Produce a software test plan for a software development project
- Design test cases that discover and isolate software defects
- Report and classify software issues and defects
- Apply effective functional (black box) testing strategies using methods such as boundary testing, equivalence classes, and decision tables
- Select and apply appropriate functional testing paradigms such as stress testing, scenario testing, specification testing, exploratory testing, etc.
- Apply effective structural (white box) testing strategies such as control flow testing and data flow testing
- Apply effective integration testing strategies such as top-down integration, bottom-up integration, and model-based integration
- Apply effective system testing strategies such as customer acceptance testing
- Achieve the goal of secure software systems with the latest security testing methods
- Understand and be able to apply the latest software testing best practices in the areas of Cloud-based Computing, Mobile Applications, and Human Usability
- Demonstrate proficiency with a state-of-the-art software testing environment (e.g. Selenium)
- Know the responsibilities of a software testing manager and the staffing requirements of an effective software testing organization
- Understand the importance of creativity in the testing of software systems

VIII. Required Course Materials

- Paul C. Jorgensen and Byron DeVries, Software Testing: A Craftsman's Approach, 5th Edition, CRC Press, Inc., Boca Raton, 2021.
- Selenium provides an open source testing environment. https://www.seleniumhq.org/download/

IX. Recommended Texts

- C. Kaner, J. Falk, and JQ Nguyen, Testing Computer Software, 2nd Edition, John Wiley & Sons, Inc., New York, 1999.
- C. Pfleeger, S. Pfleeger, and J. Margulies, Security in Computing, 5th Edition, Prentice-Hall, Inc., 2015.
- J. Whittaker, How to Break Software: A Practical Guide to Testing, Addison-Wesley, Inc., Boston, MA, 2003.
- J. Whittaker and H. Thompson, How to Break Software Security: Effective Techniques for Security Testing, Addison-Wesley, Inc., Boston, MA, 2004.

X. Basis for Final Grade

The Plus/Minus grading system will be used in this course.

Class Participation to include Online Discussion Boards	10%
Mid-Term Exam	25%
Testing Assignments	40%
• Assignment #1	10%
• Assignment #2	10%
• Assignment #3	10%
• Assignment #4	10%
Student Testing Portfolio	25%
• Written Report	20%
• Presentation	5%

There is no fixed grading scale for final grades. The instructor will determine grade cutoffs at the time of final grade assignments.

XI. Semester Project – Testing Portfolios

The goal of the semester project is for each student (or two-person team) to demonstrate their testing skills and knowledge on a software system of their choice. The project requirements are detailed in the Testing Portfolio assignment document found on Canvas.

XII. Course Schedule* (Jorgensen = Textbook, DL = Canvas and Web Downloads)

Date	Topic	Readings
February 6 *Study Sheet 1*	Introduction to Software Testing ▪ What is Software Testing? ▪ Definitions and Taxonomies of Testing Terms ▪ Testing Foundations ▪ Why is Testing So Hard? ▪ Testing Problems ▪ Mathematical Foundations	Jorgensen 1-4 DL 1-2
February 13 *Study Sheet 2*	Software Verification Techniques ▪ Software Development Life Cycle Testing ▪ Agile Testing ▪ Formal Technical Reviews ▪ Inspections ▪ Software Quality Attributes	Jorgensen 11 and 20 DL 3-5
February 20 *Study Sheet 3*	Functional Testing Methods ▪ Design of Test Cases ▪ Boundary Values ▪ Equivalence Classes ▪ Decision Tables ▪ Paradigms of Functional Testing ▪ Summary of Functional Testing	Jorgensen 5-7 Class Presentation
February 27 *Study Sheet 4*	Structural Testing Methods ▪ Control Flow Path Testing ▪ Data Flow Path Testing ▪ Coverage Metrics ▪ Object-Oriented Testing ▪ Summary of Structural Testing	Jorgensen 8-10
March 6 *Study Sheet 5*	Integration and System Testing ▪ Model-Based Testing ▪ Integration Methods ▪ System Threads	Jorgensen 12-14
March 13	Spring Break	
March 20 *Study Sheet 6*	Testing Best Practices ▪ Software Complexity ▪ Systems of Systems Testing ▪ Feature Interaction Testing ▪ Test-Driven Development (Agile Testing) ▪ The Testing Organization ▪ Managing a Testing Group Mid-Term Examination Review ▪ *Mid-Term Examination Posted on March 22, Due by the end of the day on April 1*	Jorgensen 15-18 and 21 DL 6-7
March 27 *Study Sheet 7*	Automated Testing Tools ▪ Demonstration of Selenium Tools ▪ Hands-on Laboratory with Selenium Tutorial ▪ Automated Testing Assignment	DL 8
April 3 *Study Sheet 8*	Software System Security ▪ Challenges of Software Security ▪ Secure Software Testing Methods Guest Speaker: ▪ Joe Partlow, CTO ReliaQuest	DL 9 Speaker Handouts

April 10 *Study Sheet 9*	Industry Testing Panel Guest Speakers: ▪ Bob Crews – Checkpoint Technology ▪ Deborah Sheldon – PWC ▪ Steve Splaine – Nielsen ▪ Jim Trentadue – Aristocrat Technologies	Speaker Handouts
April 17 *Study Sheet 10*	Software Program Understanding ▪ Software Behavior Computation Technology Guest Speaker: Richard Linger	DL 10-12 Speaker Handouts
April 24 *Study Sheet 11*	Student Testing Project Presentations *Testing Portfolios Due on April 28*	

* Please refer to the Course Summary & Modules within Canvas for important deadlines and a listing of what topics will be covered. Course schedule and topics are subject to change.

XIII. **Readings**

The majority of readings will be downloads available on Canvas. Other readings will provide URL links to source websites. Students should review the material on the website and download material as needed for the class discussion. A few readings come from recommended texts. All readings are to be completed by class on the date assigned.

1. Cem Kaner, "The Impossibility of Complete Testing," Law of Software Quality Column, *Software QA*, Vol. 4, No. 4, 1997.
2. James Whittaker, "What Is Software Testing? And Why Is It So Hard?" *IEEE Software*, Vol. 17, No. 1, January/February 2000.
3. B. Hungerford, A. Hevner, and R. Collins, "Reviewing Software Diagrams: A Cognitive Study," *IEEE Transactions on Software Engineering*, Vol. 30, No. 2, February 2004, pp. 82-96.
4. C. LeRouge, A. Hevner, R. Collins, M. Garfield, and D. Law, "Telemedicine Encounter Quality: Comparing Patient and Provider Perspectives of a Socio-Technical System," *Proceedings of the 37th Annual Hawaii International Conference on System Sciences* (HICSS37), Hawaii, January 2004.
5. A. Mockus, R. Fielding, and J. Herbsleb, "Two Case Studies of Open Source Software Development," *ACM Transactions on Software Engineering and Methodology*, Vol. 11, No. 3, July 2002, pp. 309-346.
6. C. Cohen, S. Birkin, M. Garfield, and H. Webb, "Managing Conflict in Software Testing," *Communications of the ACM*, Vol. 47, No. 1, January 2004.
7. Cem Kaner, Elisabeth Hendrickson, and Jennifer Smith-Brock, "Managing the Proportion of Testers to (Other) Developers," *Pacific Northwest Software Quality Conference*, Oct. 2001.
8. Selenium Open Source Testing Tools and Tutorial – Downloads in Canvas Module 7
9. G. McGraw, "Software Security," *IEEE Security & Privacy*, March/April 2004, pp. 80-83.
10. R. Collins, A. Hevner, G. Walton, and R. Linger, "The Impacts of Function Extraction Technology on Program Comprehension: A Controlled Study," *Information and Software Technology*, Vol. 50, No. 11, October 2008, pp. 1165-1179.
11. M. Pleszkoch, R. Linger, and A. Hevner, "Introducing Function Extraction into Software Testing," *The Data Base for Advances in Information Systems*, Vol. 39, No. 3, August

2008, pp. 41-50.

12. R. Linger, "Computing the Behavior of Aerospace Software for Validation of Security and Functionality," *Proceedings of 2018 AIAA Space and Astronautics Forum and Exposition,* October, 2018, Orlando, FL.

XIV. Instructor Feedback Policy & Grade Dissemination

Instructor will respond to email communication relevant to the subject matter within 12 hours of receipt. Instructor will provide feedback on assignments within one week of the posted deadline. You can access your scores at any time using "Grades" in Canvas.

XV. Course Policies

Attendance Policy: Attendance is expected for the Spring 2023 classroom course. Students are expected to take responsibility for full preparation and completion of all course requirements.

Incompletes: Only in rare cases, such as serious illness, will an Incomplete be given. An Incomplete must be requested in writing giving the reason for the request and all appropriate documentation. An Incomplete grade ("I") is exceptional and granted at the instructor's discretion only when students are unable to complete course requirements due to illness or other circumstances beyond their control. The course instructor and student must complete and sign the "I" Grade Contract Form that describes the work to be completed, the date it is due, and the grade the student would earn factoring in a zero for all incomplete assignments. The due date can be negotiated and extended by student/instructor as long as it does not exceed two semesters for undergraduate courses and one semester for graduate courses from the original date grades were due for that course. An "I" grade not cleared within the two semesters for undergraduate courses and one semester for graduate courses (including summer semester) will revert to the grade noted on the contract.

Academic Integrity: The following USF policies cover student responsibilities for issues of academic integrity:

- **Academic Integrity of Students: See**
  - http://regulationspolicies.usf.edu/regulations/pdfs/regulation-usf3.027.pdf
- **Disruption of the Academic Process: See**
  - http://regulationspolicies.usf.edu/regulations/pdfs/regulation-usf3.025.pdf
- **Student Academic Grievance Procedures: See**
  - http://regulationspolicies.usf.edu/policies-and-procedures/pdfs/policy-10-002.pdf
- **For a complete list of USF System Regulations and University Policies: See**
  - http://regulationspolicies.usf.edu/regulations-and-policies/regulations-policies-procedures.asp

XVI. USF Institutional Policies

USF Core Syllabus Policy Statements:

The following policy statements apply to all syllabi at USF and at all campuses. These policy statements are in effect even if not reproduced on the official course syllabus. https://www.usf.edu/provost/faculty-info/core-syllabus-policy-statements.aspx

USF Graduate Academic Policies:

http://www.grad.usf.edu/policies_sect7_full.php

USF Policy on University Closure

In the event of an emergency, it may be necessary for USF to suspend normal operations. During this time, USF may opt to continue delivery of instruction through methods that include but are not limited to: Canvas, MS Teams, and email messaging and/or an alternate schedule. It's the responsibility of the student to monitor the Canvas site for course specific communication, and the main USF, College, and department websites, emails, and MoBull messages for important general information.

XVII. Student Expectations

Netiquette Guidelines:

1. Act professionally in the way you communicate in email conversations or on discussion boards. Treat your instructors and peers with respect, the same way you would do in a face-to-face environment. Respect other people's ideas and be constructive when explaining your views about points you may not agree with.
2. Be sensitive. Be respectful and sensitive when sharing your ideas and opinions. There will be people in your class with different linguistic backgrounds, political and religious beliefs or other general differences.
3. Proofread and check spelling. Doing this before sending an email or posting a thread on a discussion board will allow you to make sure your message is clear and thoughtful. Avoid the use of all capital letters, it can be perceived as if you are shouting, and it is more difficult to read.
4. Keep your communications focused and stay on topic. Complete your ideas before changing the subject. By keeping the message on focus you allow the readers to easily get your idea or answers they are looking for.
5. Be clear with your message. Avoid using humor or sarcasm. Since people can't see your expressions or hear your tone of voice, meaning can be misinterpreted.

Email and Discussion Board Guidelines:

1. Use the subject line effectively by using a meaningful line of what your email or discussion is about.
2. Keep your emails and postings related to the course content. You should not post anything personal on a discussion board, unless it is requested by the instructor.

3. Any personal, course or confidential issues should be directly communicated to the instructor via email. The discussion boards are public spaces; therefore, any issues should not be posted there.

XVIII. Course Technology & Student Support

Student Accessibility Services:
Students in need of academic accommodations for a disability may consult with Student Accessibility Services (www.usf.edu/sas) to arrange appropriate accommodations. Students are required to give reasonable notice prior to requesting an accommodation. Please refer to the following guidance:

- Students with Disabilities Course Accessibility Guide: http://www.usf.edu/student-affairs/student-disabilities-services/resources/web-accessibility.aspx

- The USF Disability Accommodations Policy (#108): http://regulationspolicies.usf.edu/policies-and-procedures

- Captioning and Access of Media Policy (#10-506) at: http://regulationspolicies.usf.edu/policies-and-procedures

Academic Support Services:
The USF Office of Student Success coordinates and promotes university-wide efforts to enhance undergraduate and graduate student success. For a comprehensive list of academic support services available to all USF students, please visit the Office of Student Success website at- http://www.usf.edu/student-success/

Canvas Technical Support:
If you have technical difficulties in canvas, you can find access to the canvas guides and video resources in the “Canvas Help” page on the homepage of your canvas course. You can also contact the help desk by calling 813-974-1222 in Tampa or emailing help@usf.edu

XIX. Important Dates to Remember

University calendar dates are found at http://www.usf.edu/registrar/calendars/ .
All course dates are found on the course syllabus and in Canvas.