



The Hashemite University
Prince Al-Hussein Bin Abdullah II Faculty for Information Technology
Department of Software Engineering

Course Syllabus

Course Title: Software Testing (Master) Course Number: 121003710 Prerequisite: None Time and location: 2-3:30 (Mon, Wed)	Assessment and Course Grade: <ul style="list-style-type: none">• Midterm 25%• Homework 10%• Summaries and Presentations 10%• Term paper 15%• Final Exam 40%
Instructor: Dr. Fadi Wedyan Office No: 329 Contact Info: fadi.wedyan@hu.edu.jo	
Office Hours: Sun- Thurs: 1-2 PM, or by appointment	

Course Description

The course covers in details various aspects, theories, concepts, techniques and tools of software testing during development, maintenance and evolution. Topics include software testing at different levels (the unit, module, and system levels), testing management, inspections and walkthrough, model checking, designing and verifying test hypothesis, details of the verification and validation (V&V) techniques and concepts, bugs tracking, designing test cases and testing paths, generating of testing data. The course will be supported by practical exercises involving the development of appropriate tests and the application of a range of testing tools.

Material

- **Textbook:** Pezze & Young, Software Testing and Analysis: Process, Principles and Techniques, Wiley, 2007, ISBN: 0471455938.
- **Lecture slides:** Slides for the introductory lectures will be posted on Moodle.
- **Papers:** PDFs or links to the papers will be available in Moodle.

Additional Reading

- Paul Ammann and Jeff Offutt, Introduction to Software Testing, Cambridge University Press, Cambridge, UK, ISBN 0-52188-038-1, 2008.

Course Objectives

- Provide students with the skill to select and apply a testing strategy and testing techniques that are appropriate to a particular software system or component.
- Student will be capable of using test tools (test generation, test measurements, test running)
- Evaluate the limitations of a given testing process and provide a concise summary of those limitations

Course Plan (Tentative)

Week no.	Topic	Reading
1	Overview of software testing, Theory and Principles	Ch.1 (P & Y), Ch.4 & 20 (P & Y)
2	Functional Testing	Ch. 10 (P & Y) Ch. 4 (A & O)
3	Tools for Unit Test — JUnit	Handouts
4,5	White-box testing (Structural Testing) Introduction, Coverage Criteria, Control-flow Coverage, Data-Flow Coverage	2.1, 2.2, 2.3 (A & O)
6	Mutation Testing Hands on Tools- Coverage, Test Generation, Mutation	Handouts
Midterm Exam		
7	Testing of OO Programs	Handouts
8	Research Papers (Mutation Testing)	TBD
9	Research Papers (Mutation Testing)	TBD
10	Research Papers (Fault Localization)	TBD
11	Research Papers (Test Generation)	TBD
12	Research Papers (Testing Web Applications)	TBD
13	Research Papers (Testing Smartphone Apps)	TBD
14	Course Wrap up, Presentations	TBD
Final Exam		

Notes

- Several papers will be made available on Moodle. These papers will be selected from top journals and conferences. The papers will be discussed in class and a student will lead the discussion. Other students must also participate in the discussion. Attendance is a necessary but not sufficient condition to get participation credits for each class.

Class Schedule – Software Testing

Mutation Testing

1. Andrews, J. H., Briand, L. C., Labiche, Y., & Namin, A. S. (2006). Using mutation analysis for assessing and comparing testing coverage criteria. *Software Engineering, IEEE Transactions on*, 32(8), 608-624.
2. Siami Namin, A., Andrews, J. H., & Murdoch, D. J. (2008, May). Sufficient mutation operators for measuring test effectiveness. In *Proceedings of the 30th international conference on Software engineering* (pp. 351-360). ACM.
3. Papadakis, M., & Malevris, N. (2010, April). An empirical evaluation of the first and second order mutation testing strategies. In *Third International Conference on Software Testing, Verification, and Validation Workshops* (pp. 90-99). IEEE.
4. Just, R., & Schweiggert, F. (2015). Higher accuracy and lower run time: efficient mutation analysis using non-redundant mutation operators. *Software Testing, Verification and Reliability*, 25(5-7), 490-507.
5. Omar, E., Ghosh, S., & Whitley, D. (2017). Subtle higher order mutants. *Information and Software Technology*, 81, 3-18.

Testing Web, GUI, and Mobile Applications

6. Deng, L., Offutt, J., Ammann, P., & Mirzaei, N. (2017). Mutation operators for testing Android apps. *Information and Software Technology*, 81, 154-168.
7. Jabbarvand, R., & Malek, S. (2017, August). μ Droid: an energy-aware mutation testing framework for Android. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 208-219). ACM.
8. Moran, K., Tufano, M., Bernal-Cárdenas, C., Linares-Vásquez, M., Bavota, G., Vendome, C., ... & Poshyvanyk, D. (2018). MDroid+: A Mutation Testing Framework for Android. Accepted to the Formal Tool Demonstration Track at the 40th International Conference on Software Engineering (ICSE'18)
9. Alégroth, E., Feldt, R., & Kolström, P. (2016). Maintenance of automated test suites in industry: An empirical study on Visual GUI Testing. *Information and Software Technology*, 73, 66-80.
10. Mesbah, A., Van Deursen, A., & Roest, D. (2012). Invariant-based automatic testing of modern web applications. *Software Engineering, IEEE Transactions on*, 38(1), 35-53.
11. Mukherjee, J., Wang, M., & Krishnamurthy, D. (2014, March). Performance Testing Web Applications on the Cloud. In *Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference on* (pp. 363-369). IEEE.
12. Delamaro, M. E., Vincenzi, A. M. R., & Maldonado, J. C. (2006, May). A strategy to perform coverage testing of mobile applications. In *Proceedings of the 2006 international workshop on Automation of software test* (pp. 118-124). ACM.
13. Yang, W., Prasad, M. R., & Xie, T. (2013). A grey-box approach for automated GUI-model generation of mobile applications. In *Fundamental Approaches to Software Engineering* (pp. 250-265). Springer Berlin Heidelberg.

14. Amalfitano, D., Fasolino, A. R., & Tramontana, P. (2011, March). A gui crawling-based technique for android mobile application testing. In *Software Testing, Verification and Validation Workshops (ICSTW), 2011 IEEE Fourth International Conference on* (pp. 252-261). IEEE.
15. Amalfitano, D., Fasolino, A. R., Tramontana, P., De Carmine, S., & Memon, A. M. (2012, September). Using GUI ripping for automated testing of Android applications. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering* (pp. 258-261). ACM.
16. Jensen, C. S., Prasad, M. R., & Møller, A. (2013, July). Automated testing with targeted event sequence generation. In *Proceedings of the 2013 International Symposium on Software Testing and Analysis* (pp. 67-77). ACM.
17. Nguyen, B. N., Robbins, B., Banerjee, I., & Memon, A. (2014). GUITAR: an innovative tool for automated testing of GUI-driven software. *Automated Software Engineering*, 21(1), 65-105.
18. Lelli, V., Blouin, A., & Baudry, B. (2015, April). Classifying and qualifying GUI defects. In *Software Testing, Verification and Validation (ICST), 2015 IEEE 8th International Conference on* (pp. 1-10). IEEE.
19. McMinn, P., Wright, C. J., McCurdy, C. J., & Kapfhammer, G. (2017). Automatic Detection and Removal of Ineffective Mutants for the Mutation Analysis of Relational Database Schemas. *IEEE Transactions on Software Engineering*.