

Jeff Offutt, PhD

Professor
Software Engineering
Volgenau School of Engineering George Mason University
Fairfax, VA 22030-4444

offutt@gmu.edu
www.cs.gmu.edu/~offutt/

RESEARCH, TEACHING, AND BRIEF BIO

Jeff Offutt has published over 180 refereed research papers, has an h-index of 61 (Google Scholar), and has received funding from many government agencies and companies. His current research include leading the SPARC (Self-Paced Learning Increases Retention and Capacity) project at George Mason, the Testing of Critical System Characteristics (TOCSYC) and PILOT projects at University of Skövde, and smaller projects in mobile applications, analysis and testing of web applications, model-based testing, test automation, and usable security. He was on the technical board of advisors for Certess, Inc. Offutt was awarded the George Mason University Teaching Excellence Award (Teaching With Technology) in 2013. He leads the MS in Software Engineering program at GMU, teaches Software Engineering courses at all levels and has developed new courses on several Software Engineering subjects, including web engineering, software testing, construction, design, usability, experimentation, and analysis. His textbook, *Introduction to Software Testing* (co-authored with Paul Ammann), is the leading worldwide textbook in software testing and the second edition was published in 2016. Offutt was named a George Mason University Outstanding Faculty member in 2008 and 2009. Offutt is editor-in-chief of Wiley's journal of Software Testing, Verification and Reliability, co-founded the IEEE International Conference on Software Testing, Verification and Validation (ICST), was the first steering committee chair, and was Program Chair for ICST 2009. He also has served on dozens of conference program committees and been on the editorial boards for the Springer's Empirical Software Engineering Journal (2006-), the Journal of Software and Systems Modeling (2002-), the Software Quality Journal (2002-), and IEEE Transactions on Software Engineering (2001-2005), is a regular reviewer for NSF and several major research journals, and has been invited to speak throughout the world. He has been involved in a number of software proof-of-concept research systems, including muJava, coverage analysis web apps, Mothra, Godzilla, CBat, Mistix, Albert, CoupTest, and SpecTest, which have been used by thousands of other software engineering researchers. He has invented, developed, and experimentally validated numerous algorithms and engineering techniques in software testing, many of which are in widespread use. Offutt has made fundamental contributions to several software testing problems, including mutation, specification-based testing, model-based testing, automatic test data generation, object-oriented testing, input space partitioning, and testing of web applications. He has also published on software metrics, maintenance, and software engineering education.

PROFESSIONAL EXPERIENCE

- **since 2005:** Full Professor (with Tenure) of Software Engineering, George Mason University. Assistant Professor 1992–1996, Associate Professor with tenure 1996-2005.
- **Part-time and Visiting Positions:**
 - **since 2002:** Part-time visiting professor in the Department of Computer Science, University of Skövde, Skövde Sweden.
 - **2010-2013:** Part-time visiting professor in the Department of Computer Science, University of Linköping, Linköping Sweden.
 - **January 2000 to August 2007:** Part-time Research Scientist with the National Institute of Standards and Technology's Information Technology Lab.
- **January 1998 to August 1998:** Acting Chair of Information and Software Engineering Department, George Mason University.
- **August 1988 to August 1992:** Assistant Professor of Computer Science, Clemson University.
- **January 1983 to August 1988:** Graduate Research and Teaching Assistant in the Software Engineering Research Center and in the Department of Information and Computer Science, Georgia Institute of Technology.

EDUCATION

- PhD Information and Computer Science, Georgia Institute of Technology, August 1988.
- MS Information and Computer Science, Georgia Institute of Technology, March 1985.
- BS cum laude Mathematics and Data Processing (double major), minor in Physics, Morehead State University, May 1982.

GRANTS

- *Preparing K-5 Teachers to Integrate the Computer Science Standards of Learning in Inclusive Classrooms to Support Students with High Incidence Disabilities*, (lead PI: Amy Hutchison, GMU, joint with GMU and ODU), NSF Computer Science for All progra, \$999,423, October 2018 - September 2021.
- *SPARC: Self-Paced Learning increases Retention and Capacity*, Google Education Grant, 3X in 3 Years Awards program, \$900,000, February 2015 - February 2018.
- *Usable Analysis of Security Protocols*, Department of Homeland Security, \$128,993, February 2012 - February 2014.
- *Testing of Critical System Characteristics (TOCSYC)*, Swedish Knowledge Foundation, 30,972,948 SEK (~\$4,700,000 USD), 2013-2018 (Primary: Mälardalen University, subcontracted to Skövde University).
- *PILOT*, Swedish Vinnova and Saab Aeronautics, 6,500,000 SEK (~\$1,000,000 USD), 2013-2015 (Primary: Skövde University).
- *Research into Testing Service Oriented Architectures*, Avaya Research Labs, \$61,169, July 2006 - May 2007.
- *Coupling-based Object-Oriented Software Analysis*, NIST Information Technology Lab, \$24,988, June 2005 - January 2006.
Coupling-based Analysis of Object-Oriented Software and Web Services, \$24,990, March 2006 - August 2006.
- *Assuring Web-based Software System Components*, NASA Goddard, subcontract through Indus Corporation: \$104,331, September 2003 - December 2004.
- *Repeated Maintenance of Open-Source Software*, NSF: CCR - 00 97056, \$225,000, Sept 2001 - December 2004. Collaborative with Steven Schach of Vanderbilt University.
REU Supplemental Research Experience for Undergraduates: \$6000, June 2002 - Sept 2002.
REU Supplemental Research Experience for Undergraduates: \$6459, June 2003 - Sept 2003.
- *Coupling-based Analysis for Integration Testing of Object-oriented Software*, NSF: CCR - 98 04111, \$200,000, July 1998 - June 2001.
- *A Comparative Evaluation of Data Flow and Mutation Testing*, NSF: CCR - 93 11967, \$107,390, August 1993 - January 1996.
- *Assuring Web-based Software System Components*, NASA Software IV&V Facility: \$136,000, 2001. (Note: This grant was awarded but funding was not released.)
- *Generating Test Cases From Requirements/Specifications*, Rockwell-Collins Avionics
Phase I: \$25,000, May 1997 - August 1997.
Phase II: \$40,000, January 1998 - August 1998.
Phase III: \$30,000, January 1999 - August 1999.
Phase IV: \$30,000, January 2000 - August 2000.
- *Estimator Efficiency*, Software Productivity Consortium, \$29,200, January 1999 - May 1999.
- *Software Reliability for Real-Time Control Systems*, NASA Langley Research Center, \$20,000, August 1989 - August 1990.

- *A Practical Mutation Testing System for Ada*, NASA Phase I SBIR grant to Reliable Software Technologies Corporation, February 1993 - July 1993 (primary author but listed as consultant, Jeff Voas was PI).
- *Specification-based testing*, 1999-2001, The Ministry of Education of Japan under Joint Research Grant-in-Aid for International Scientific Research FM-ISEE (08044167). PI is Dr. Shaoying Liu of Hosei University, Tokyo Japan, grant supported collaboration by funding several trips.
- *XML Testing of Web Services*, IT&E Graduate Research Assistantship, August 2005 - May 2006.
- *Testing Data State Interactions Web Software Applications*, IT&E Graduate Research Award, August 2002 - May 2003.
- *Software Testing Based on Module Coupling*, IT&E Graduate Research Assistantship, August 1995 - May 1996.

CONSULTANT ACTIVITIES

- Various Software Intellectual Property Cases: TransUnion, TriZetto, National Wooden Pallet and Container Association, Vertel, Creative Labs, Agitar Software
- Technical Consulting and Advising: Hyperchip, Inc., IP Optical, Inc., Certess Inc (member of the Technical Advisory Board), Bell Communications Research, International Research Institute, Reliable Software Technologies, Rockwell Collins Avionics
- Educational Consulting and Advising: Laureate, Samsung Electronics, United Arab Emirates University

Author Impact Analysis

h-index: 61; Number of total citations: 17,206; Citations per year: 593 (Source: Google Scholar)

Honors and Awards

- Outstanding Paper Award, Mutation Workshop 2018
- "Putting the Engineering into Software Engineering Education," IEEE Software 30(1), February 2013, was selected by ACM Computing Reviews as a *notable article* for 2013
- George Mason University Teaching Excellence Award, Teaching With Technology, 2013
- Finalist, Governor's Technology in Education Award, 2012
- GMU Outstanding Faculty member, 2009, 2010
- Outstanding Researcher Award, ISE Department, 2006
- Outstanding Researcher Award, School of IT&E, 2004
- Best Teacher Award, School of IT&E, 2003
- Outstanding Teacher Award, ISE Department, 2003
- Outstanding Paper Award, ICECCS 1996

INVENTIONS AND INNOVATIONS

Offutt and his collaborators have contributed numerous inventions and innovations in software engineering, most significantly in software testing.

- Mutation testing: Offutt has contributed dozens of major results to this topic, collectively resulting in a comprehensive engineering solution to make mutation practical for industrial use. Many of these results are incorporated into the commercial tool Certess, by Certitude Inc.
 - Developed two tools for widespread use; Mothra was used throughout the testing community in the 1990s and 1980s, muJava is currently used by hundreds of researchers and students (with former student Dr. Yu-Seung Ma)

- Empirically verified the coupling effect, a basic premise for mutation
- Invented the schema-based approach for applying mutation, which solves most of the performance problems (with former student Dr. Roland Untch)
- Invented techniques to detect equivalent mutants, greatly reducing a huge bottleneck to the adoption of mutation (with former student Jie Pan)
- Empirically demonstrated that weak mutation is a viable alternative to strong mutation (with former student Stephen Lee)
- Developed several processes for how best to apply mutation
- Invented algorithms for parallelizing mutation
- Empirically showed that mutation is superior to several other testing techniques, most notably data flow
- Developed mutation operators for real-time testing (with former student Dr. Robert Nilsson)
- Automatic test data generation: Offutt has contributed several results to this difficult topic in which most of the problems are generally undecidable. Many of these results are incorporated into commercial tools such as Agitator and Pex
 - Invented algorithms for generating test data that satisfy statement coverage, branch coverage, data flow coverage and mutation coverage for unit testing
 - Invented heuristics for recognizing many infeasible test requirements (generally undecidable)
 - Invented the dynamic domain reduction procedure, currently the strongest algorithm for generating test data
- Developed new methods and processes for applying input space partitioning (with former student Dr. Mats Grindal)
- Developed empirically techniques for evaluating maintainability of software (with Dr. Steve Schach at Vanderbilt University)
- Developed metrics for use with object-oriented software, coupling, and component-based software
- Developed several concepts for testing web applications and services, including data perturbation, bypass testing, finite state machine modeling, and atomic section modeling (with Dr. Ye Wu, Dr. Anneliese Andrews and current student Wuzhi Xu)
- Invented prime path testing (with Dr. Paul Ammann)
- Invented base choice testing (with Dr. Paul Ammann)
- Invented coupling-based testing for object-oriented software
- Invented models and algorithms for testing inheritance and polymorphism relations in object-oriented software (with former student Dr. Roger Alexander)
- Developed input validation testing, based on requirements documents (with former student Dr. Jane Hayes)
- Invented the first testing techniques based on UML diagrams, including new criteria; this work helped spawn a new sub-field in testing called “model-based testing” (with former student Dr. Aynur Abdurazik)
- Developed new algorithms for solving the class integration and test order problem (with former student Dr. Aynur Abdurazik)

PUBLICATIONS

● BOOKS AND CHAPTERS

1. Paul Ammann and **Jeff Offutt**, *Introduction to Software Testing*, second edition, 2016, Cambridge University Press (first edition 2008).
2. “Coverage Criteria for State Based Specifications,” Paul Ammann, Jeff Offutt and Wuzhi Xu, chapter in “Formal Methods and Testing,” edited by Rob Hierons, Jonathan Bowen and Mark Harman, Springer-Verlag Lecture Notes in Computer Science 4949, pages 118-156, April 2008. DOI 10.1007/978-3-540-78917-8. (*Invited*).

3. “Software Design and Implementation in the Web Environment,” chapter in “The Internet Encyclopedia,” edited by Hossein Bidgoli, John Wiley & Sons, Inc. 2003.

- REFEREED JOURNAL PUBLICATIONS

(My students’ names are *highlighted*.)

1. **Yun Guo**, Nan Li, Jeff Offutt and Ami Motro. Exoneration-based Fault Localization for SQL Predicates. Elsevier’s *Journal of Systems and Software*, accepted per revision, 2018.
2. Jeff Offutt and **Sunitha Thummala**. Testing Concurrent User Behavior of Synchronous Web Applications with Petri Nets. Springer’s *Software and Systems Modeling*, online February 2018.
3. Vinicius H. S. Durelli, Marcio E. Delamaro, and Jeff Offutt. An Experimental Comparison of Edge, Edge-Pair, and Prime Path Criteria. Elsevier’s *Science of Computer Programming*, 152(15):99-115, January 2018.
4. **Nan Li** and Jeff Offutt. Test Oracle Strategies for Model-based Testing. *IEEE Transactions on Software Engineering*, 43(4):372-395, April 2017.
5. **Lin Deng**, Jeff Offutt, Paul Ammann, and Nariman Mirzaei. Mutation Operators for Testing Android Apps. Elsevier’s *Information and Software Technology, special issue from the mutation 2015 workshop*, 81:154-168, January 2017.
6. Birgitta Lindström, Jeff Offutt, Daniel Sundmark, Sten F. Andler, Paul Pettersson. Using Mutation to Design Tests for Aspect-Oriented Models. Elsevier’s *Information and Software Technology, special issue from the mutation 2015 workshop*, 81:154-168, January 2017.
7. Deanna D. Caputo, Shari Lawrence Pfleeger, M. Angela Sasse, Paul Ammann, Jeff Offutt, and **Lin Deng**. Barriers to Usable Security? Three Organizational Case Studies. *IEEE Security & Privacy*, 14(5):22-32, September-October 2016.
8. Vinicius H. S. Durelli, Jeff Offutt, **Nan Li**, Marcio E. Delamaro, Jin Guo, Zengshu Shi, Xinge Ai. What to Expect of Predicates: An Empirical Analysis of Predicates in Real World Programs. Elsevier’s *Journal of Systems and Software*, vol. 113:324-336, March 2016. DOI: 10.1016/j.jss.2015.12.022
9. Mark Ardis, David Budgen, Gregory W. Hislop, Jeff Offutt, Mark Sebern, and Willem Visser. SE2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. *IEEE Computer*, 48(11):106-109, November 2015. Full report online: <http://www.acm.org/education/se2014.pdf>.
10. Jeff Offutt and **Chandra Alluri**. An Industrial Study of Applying Input Space Partitioning to Test Financial Calculation Engines. Springer’s *Empirical Software Engineering journal*, 19(3):558-581, June 2014.
11. Jeff Offutt, **Vasileios Papadimitriou**, and **Upsorn Praphamontripong**. A Case Study on Bypass Testing of Web Applications. Springer’s *Empirical Software Engineering journal*, 19(1):69-104, February 2014.
12. Jeff Offutt. Putting the Engineering into Software Engineering Education. *IEEE Software*, Jan-Feb 2013, 30(1):96-100. (*Opinion column*.)
13. Gary Kaminski, Paul Ammann, Jeff Offutt. Improving Logic-Based Testing. Elsevier’s *Journal of Systems and Software*, 86(9):2002-2012, August 2013. DOI 10.1016/j.jss.2012.08.024,
14. Pedro Reales Mateo, Macario Polo Usaola, and Jeff Offutt. Mutation at the Multi-class and System Levels. Elsevier’s *Science of Computer Programming*, 78(4):364-387, April 2013.
15. Jeff Offutt. A Mutation Carol: Past, Present and Future. Elsevier’s *Information and Software Technology, special issue from the mutation 2009 workshop*, 53(10):1098-1107, October 2011.
16. **Garrett Kaminski**, **Upsorn Praphamontripong**, Paul Ammann, Jeff Offutt. A Logic Mutation Approach to Selective Mutation for Programs and Queries. Elsevier’s *Information and Software Technology, special issue from the mutation 2009 workshop*, 53(10):1137-1152, October 2011.
17. **Roger T. Alexander**, Jeff Offutt, and Andreas Stefik. Testing Coupling Relationships in Object-Oriented Programs. Wiley’s *Journal of Software Testing, Verification, and Reliability*, 20(4):291-327, December 2010.
18. **Jane Hayes** and Jeff Offutt. Recognizing Authors: An Examination of the Consistent Programmer Hypothesis. Wiley’s *Journal of Software Testing, Verification, and Reliability*, 20(4):329-356, December 2010.

19. Anneliese A. Andrews, Jeff Offutt, Curtis Dyreson, Christopher J. Mallery, Kshamta Jerath, and **Roger Alexander**. Scalability Issues with Using FSMWeb to Test Web Applications. Elsevier's *Information and Software Technology*, 52(1):52-66, January 2010 (DOI: 10.1016/j.infsof.2009.06.002).
20. Jeff Offutt and Ye Wu. Modeling Presentation Layers of Web Applications for Testing. Springer's *Software and Systems Modeling*, 9(2):257-280, April 2010.
21. Larry G. Thomas, Stephen R. Schach, Gillian Z. Heller, Jeff Offutt. Impact of Release Intervals on Empirical Research into Software Evolution, with Application to the Maintainability of Linux. *IET Software*, 3(1):58-66, February 2009.
22. **Aynur Abdurazik** and Jeff Offutt. Using Coupling-based Weights for the Class Integration and Test Order Problem. *The Computer Journal*, pages 1-14, August 2009.
23. Leonard Gallagher and Jeff Offutt. Test Sequence Generation for Integration Testing of Component Software. *The Computer Journal*, 52(5):514-529, August 2009, doi: 10.1093/comjnl/bxm093.
24. Jeff Offutt, **Aynur Abdurazik** and Steve Schach. Quantitatively Measuring Object-Oriented Couplings. Springer's *Software Quality Journal*, 6(4):489-517, December 2008, doi: 10.1007/s11219-008-9051-x.
25. Leonard Gallagher, Jeff Offutt, and Anthony Cincotta. Integration Testing of Object-oriented Components Using Finite State Machines. *Journal of Software Testing, Verification, and Reliability*, Wiley, 17(1):215-266, January 2007.
26. **Supaporn Kansomkeat**, Jeff Offutt, and Wanchai Rivepiboon. Bytecode-based Analysis for Increasing Class-Component Testability. *ECTI Transactions on Computer and Information Technology*, 2(2):33-44, November 2006.
27. **Mats Grindal**, **Birgitta Lindström**, Jeff Offutt, and Sten F. Andler. An Evaluation of Combination Testing Strategies. *Empirical Software Engineering*, 11(4):583-611, December 2006.
28. **Jane Hayes** and Jeff Offutt. Input Validation Analysis and Testing. *Empirical Software Engineering*, 11(4):493-522, December 2006.
29. Ligu Yu, Stephen R. Schach, Kai Chen, Gillian Z. Heller, and Jeff Offutt. Maintainability of the Kernels of Open-Source Operating Systems: A Comparison of Linux with FreeBSD, NetBSD, and OpenBSD. *Journal of Systems and Software*, 79:807-815, December 2005.
30. **Mats Grindal**, Jeff Offutt, and Sten F. Andler. Combination Testing Strategies: A Survey. *Journal of Software Testing, Verification and Reliability*, Wiley, 15(2):97-133, September 2005.
31. **Yu-Seung Ma**, Jeff Offutt, and Yong Rae Kwon. MuJava : An Automated Class Mutation System. *Journal of Software Testing, Verification and Reliability*, Wiley, 15(2)97-133, June 2005.
32. Anneliese Andrews, Jeff Offutt, and **Roger Alexander**. Testing Web Applications by Modeling with FSMs. *Software Systems and Modeling*, 4(3):326-345, July 2005.
33. Ligu Yu, Stephen R. Schach, Kai Chen, and Jeff Offutt. Categorization of Common Coupling and its Application to the Maintainability of the Linux Kernel. *IEEE Transactions on Software Engineering*, 30(10):694-706, October 2004.
34. **Roger Alexander** and Jeff Offutt. Coupling-based Testing of OO Programs. Springer's *Journal of Universal Computer Science: Special Issue on Breakthroughs and Challenges in Software Engineering* (invited), 10(4):391-427, April 2004.
35. Kai Chen, Stephen R. Schach, Ligu Yu, and Jeff Offutt. Open-Source Change Logs. Kluwer's *Empirical Software Engineering*, 9(3):197-210, September 2004.
36. Steve Schach, Bo Jin, Ligu Yu, Gillian Z. Heller, and Jeff Offutt. Determining the Distribution of Maintenance Categories: Survey versus Measurement. *Kluwer's Empirical Software Engineering*, 8(4):351-365, December 2003.
37. Steve Schach, Bo Jin, David R. Wright, Gillian Z. Heller, and Jeff Offutt. Quality Impacts of Clandestine Common Coupling. *Kluwer's Software Quality Journal*, 11(3):211-218, July 2003.
38. Jeff Offutt, Shaoying Liu, **Aynur Abdurazik**, and Paul Ammann. Generating Test Data from State-based Specifications. *The Journal of Software Testing, Verification, and Reliability*, Wiley, 13(1):25-53, March 2003.

39. Steve Schach, Bo Jin, David Wright, Gillian Z. Heller, and Jeff Offutt. Maintainability of the Linux Kernel. *IEE Proceedings Journal: Special Issue on Open Source Software Engineering*, 149(1):18–23, February 2002.
 40. Jeff Offutt. Quality Attributes of Web Software Applications. *IEEE Software: Special Issue on Software Engineering of Internet Software*, 19(2):25–32, March/April 2002.
 41. Jeff Offutt and Shaoying Liu. Generating Test Data from SOFL Specifications. *The Journal of Systems and Software*, 49(1):49–62, December 1999.
 42. Jeff Offutt, **Zhenyi Jin**, and **Jie Pan**. The Dynamic Domain Reduction Procedure for Test Data Generation. *Software Practice and Experience*, 29(2):167–193, January 1999.
 43. Jeff Offutt and **Zhenyi Jin**. Coupling-based Criteria for Integration Testing. *The Journal of Software Testing, Verification, and Reliability*, Wiley, 8(3):133–154, September 1998.
 44. Shaoying Liu, Jeff Offutt, Mitsuru Ohba, and Keijiro Araki. The SOFL Approach: An Improved Principle for Requirements Analysis. *Transactions of Information Processing Society of Japan*, 39(6):1973–1989, June 1998.
 45. Shaoying Liu, Jeff Offutt, Chris Ho-Stuart, Yong Sun, and Mitsuru Ohba. SOFL : A Formal Engineering Methodology for Industrial Applications. *IEEE Transactions on Software Engineering*, Special Issue on Formal Methods, 24(1):337–344, January 1998.
 46. Jeff Offutt and **Jie Pan**. Automatically Detecting Equivalent Mutants and Infeasible Paths. *The Journal of Software Testing, Verification, and Reliability*, Wiley, 7(3):165–192, September 1997.
 47. Mary Jean Harrold, Jeff Offutt, and **Kanupriya Tewary**. An Approach to Fault Modeling and Fault Seeding using the Program Dependence Graph. *The Journal of Systems and Software*. 36(3):273–296, March 1997.
 48. Jeff Offutt, Gregg Rothermel, **Christian Zapf**, **Roland Untch**, and **Ammei Lee**. An Experimental Determination of Sufficient Mutation Operators. *ACM Transactions on Software Engineering Methodology*. 5(2):99–118, April 1996.
 49. Jeff Offutt, **Jie Pan**, **Tong Zhang**, and **Kanupriya Tewary**. An Experimental Evaluation of Data Flow and Mutation Testing. *Software Practice and Experience*, 26(2):165–176, February 1996.
 50. Jeff Offutt and **W. M. Craft**. Using Compiler Optimization Techniques to Detect Equivalent Mutants. *The Journal of Software Testing, Verification, and Reliability*, Wiley, 4(3):131–154, September 1994.
 51. Jeff Offutt and **Stephen D. Lee**. An Empirical Evaluation of Weak Mutation. *IEEE Transactions on Software Engineering*, 20(5):337–344, May 1994.
 52. Rich DeMillo and Jeff Offutt. Experimental Results from an Automatic Test Case Generator. *ACM Transactions on Software Engineering Methodology*, 2(2):109–175, April 1993.
 53. Jeff Offutt, Mary Jean Harrold, and P. Kolte. A Software Metric System for Module Coupling. *The Journal of Systems and Software*, 20(3):295–308, March 1993.
 54. Robert Geist, Jeff Offutt, and Fred Harris. Estimation and Enhancement of Real-Time Software Reliability through Mutation Analysis. *IEEE Transactions on Computers Special Issue on Fault-Tolerant Computing*, 41(5):550–558, May 1992.
 55. Jeff Offutt. Investigations of the Software Testing Coupling Effect. *ACM Transactions on Software Engineering Methodology*, 1(1):3–18, January 1992.
 56. Jeff Offutt. An Integrated Automatic Test Data Generation System. *Journal of Systems Integration*, 1(3):391–409, November 1991.
 57. Rich DeMillo and Jeff Offutt. Constraint-Based Automatic Test Data Generation, *IEEE Transactions on Software Engineering*, 17(9):900–910, September 1991.
 58. K. N. King and Jeff Offutt. A Fortran Language System for Mutation-Based Software Testing. *Software Practice and Experience*, 21(7):686–718, July 1991.
 59. Jeff Offutt. Software Testing Technology. *The ITEA Journal of Test and Evaluation*, 7(2):18–31, Spring 1986.
- REFEREED CONFERENCE PUBLICATIONS
(My students' names are **highlighted**.)

60. **Yun Guo**, Nan Li, Jeff Offutt, and Ami Motro. Automatically Repairing SQL Faults. 18th IEEE International Conference on Software Quality, Reliability, and Security (QRS), Lisbon Portugal, July 2018.
61. Loreto Gonzalez-Hernandez, **Birgitta Lindström**, Jeff Offutt, Sten Andler, Pasqualina Potena, and Markus Bohlin. Using Mutant Stubbornness to Create Minimal and Prioritized Test Sets. 18th IEEE International Conference on Software Quality, Reliability, and Security (QRS), Lisbon Portugal, July 2018.
62. **Lin Deng** and Jeff Offutt. Reducing the Cost of Android Mutation Testing. International Conference on Software Engineering and Knowledge Engineering (SEKE 2018), San Francisco CA, USA, July 2018.
63. **Ryan Johnson**, M. ElSabagh, Angelos Stavrou, and Jeff Offutt. Dazed Droids: A Longitudinal Study of Android Inter-App Vulnerabilities. 13th ACM ASIA Conference on Information, Computer and Communications Security (ACM ASIACCS 2018), Incheon Korea.
64. Fabiano Cutigi Ferrari, Alessandro Viola Pizzoleto, and Jeff Offutt. A Systematic Review of Cost Reduction Techniques for Mutation Testing: Preliminary Results. *Fourteenth IEEE Workshop on Mutation Analysis (Mutation 2018)*, April 2018, Vasteros, Sweden. (Best paper award.)
65. **Birgitta Lindström**, Jeff Offutt, Loreto Gonzalez-Hernandez, and Sten Andler. Identifying Useful Mutants to Test Time Properties. 2nd International Workshop on Testing Extra-Functional Properties and Quality Characteristics of Software Systems (ITEQS), April 2018, Vasteros, Sweden.
66. **Lin Deng**, Jeff Offutt, and David Samudio. Is Mutation Analysis Effective at Testing Android Apps? *IEEE International Conference on Software Quality, Reliability and Security (QRS)*, July 2017, Prague, Czech Republic.
67. **Upsorn Praphamontripong** and Jeff Offutt. Finding Redundancy in Web Mutation Operators. *Thirteenth IEEE Workshop on Mutation Analysis (Mutation 2017)*, April 2017, Tokyo, Japan.
68. Jeff Offutt, Paul Ammann, Kinga Dobolyi, Chris Kauffman, Jaime Lester, **Upsorn Praphamontripong**, Huzefa Rangwala, Sanjeev Setia, Pearl Wang, and Liz White. A Novel Self-Paced Model for Teaching CS1 and CS2. Learning at Scale, April 2017, Boston, USA.
69. **Bob Kurtz**, Paul Ammann, Jeff Offutt, Marcio E. Delamaro, Mariet Kurtz, and Nida Gökçe. Analyzing the Validity of Selective Mutation with Dominator Mutants. *24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, November 2016, Seattle Washington, USA.
70. **Bob Kurtz**, Paul Ammann, Jeff Offutt, and Mariet Kurtz. Are We There Yet? How Redundant and Equivalent Mutants Affect Determination of Test Completeness. *Twelfth IEEE Workshop on Mutation Analysis (Mutation 2016)*, April 2016, Chicago Illinois, USA.
71. **Upsorn Praphamontripong**, Jeff Offutt, **Lin Deng**, and **JingJing Gu**. An Experimental Evaluation of Web Mutation Operators. *Twelfth IEEE Workshop on Mutation Analysis (Mutation 2016)*, April 2016, Chicago Illinois, USA.
72. **Sunitha Thummala** and Jeff Offutt. Using Petri Nets to Test Concurrent Behavior of Web Applications. 12th Workshop on Advances in Model-based testing (A-MOST 2016), April 2016, Chicago Illinois, USA.
73. **Nan Li**, Anthony Escalona, Yun Guo, and Jeff Offutt. A Scalable Big Data Test Framework. 8th IEEE International Conference on Software Testing, Verification and Validation (ICST), Testing in Practice, April 2015, Graz, Austria.
74. **Birgitta Lindström**, Sten Andler, Jeff Offutt, Paul Pettersson, and Daniel Sundmark. Mutating Aspect-Oriented Models to Test Cross-Cutting Concerns. *Eleventh IEEE Workshop on Mutation Analysis (Mutation 2015)*, April 2015, Graz, Austria.
75. **Bob Kurtz**, Paul Ammann, and Jeff Offutt. Static Analysis of Mutant Subsumption. *Eleventh IEEE Workshop on Mutation Analysis (Mutation 2015)*, April 2015, Graz, Austria.
76. **Jing Guan** and Jeff Offutt. A Model-Based Testing Technique for Component-Based Real-Time Embedded Systems. 11th Workshop on Advances in Model-based testing (A-MOST 2015), April 2015, Graz, Austria.
77. **Nan Li** and Jeff Offutt. A Test Automation Language Framework for Behavioral Models. 11th Workshop on Advances in Model-based testing (A-MOST 2015), April 2015, Graz, Austria.

78. **Lin Deng**, Nariman Mirzaei, Paul Ammann, and Jeff Offutt. Towards Mutation Analysis of Android Apps. *Eleventh IEEE Workshop on Mutation Analysis (Mutation 2015)*, April 2015, Graz, Austria.
79. **Sunitha Thummala** and Jeff Offutt. An Evaluation of the Effectiveness of the Atomic Section Model. ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014), September 2014, Valencia Spain
80. **Bob Kurtz**, Paul Ammann, Marcio E. Delamaro, Jeff Offutt, and **Lin Deng**. Mutant Subsumption Graphs. *Tenth Workshop on Mutation Analysis (Mutation 2014)*, March 2014, Cleveland Ohio, USA.
81. Marcio E. Delamaro and Jeff Offutt. Assessing the Influence of Multiple Test Case Selection on Mutation Experiments. *Tenth Workshop on Mutation Analysis (Mutation 2014)*, March 2014, Cleveland Ohio, USA.
82. Marcio E. Delamaro, Jeff Offutt, and Paul Ammann. Designing Deletion Mutation Operators. *7th IEEE International Conference on Software Testing, Verification and Validation (ICST 2014)*, March 2014, Cleveland Ohio, USA.
83. Paul Ammann, Marcio E. Delamaro, and Jeff Offutt. Establishing Theoretical Minimal Sets of Mutants. *7th IEEE International Conference on Software Testing, Verification and Validation (ICST 2014)*, March 2014, Cleveland Ohio, USA.
84. Marcio E. Delamaro, **Lin Deng**, Vinicius H. S. Durelli, **Nan Li**, and Jeff Offutt. Experimental Evaluation of SDL and One-Op Mutation for C. *7th IEEE International Conference on Software Testing, Verification and Validation (ICST 2014)*, March 2014, Cleveland Ohio, USA.
85. **Nan Li** and Jeff Offutt. An Empirical Analysis of Test Oracle Strategies for Model-based Testing. *7th IEEE International Conference on Software Testing, Verification and Validation (ICST 2014)*, March 2014, Cleveland Ohio, USA.
86. Marcio E. Delamaro, **Lin Deng**, **Nan Li**, Vinicius H. S. Durelli, and Jeff Offutt. Growing a Reduced Set of Mutation Operators. 2014 IEEE Brazilian Symposium on Software Engineering.
87. **Nan Li**, **Xin Meng**, Jeff Offutt, and **Lin Deng**. Is Bytecode Instrumentation as Good as Source Instrumentation: An Empirical Study with Industrial Tools. *24th IEEE International Symposium on Software Reliability Engineering (ISSRE 2013)*, pages 380-389, November 2013, Pasadena California, USA.
88. **Jing Jin**, Jeff Offutt, Nan Zheng, Feng Mao, Aaron Koehl, and Haining Wang. Evasive Robots Masquerading as Human Beings on the Web. *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DNS 2013)*, June 2013, Budapest Hungary.
89. **Anders Eriksson**, **Birgitta Lindström**, and Jeff Offutt. Transformation Rules for Platform Independent Testing: An Empirical Study. *6th IEEE International Conference on Software Testing, Verification and Validation (ICST 2013)*, March 2013, Luxembourg.
90. **Lin Deng**, Jeff Offutt, and **Nan Li**. Empirical Evaluation of the Statement Deletion Mutation Operator. *6th IEEE International Conference on Software Testing, Verification and Validation (ICST 2013)*, March 2013, Luxembourg.
91. **Anders Eriksson**, **Birgitta Lindström**, Sten Andler, and Jeff Offutt. Model Transformation Impact on Test Artifacts: An Empirical Study. *9th Model-Driven Engineering, Verification, and Validation: Integrating Verification and Validation in MDE (MoDeVVA 2012)*, Sept 2012, Innsbruck, Austria.
92. Vinicius Durelli, Jeff Offutt, and Marcio Delamaro. Toward Harnessing High-level Language Virtual Machines for Further Speeding up Weak Mutation Testing. *Eighth Workshop on Mutation Analysis (Mutation 2012)*, April 2012, Montreal, Canada.
93. William Shelton, **Nan Li**, Paul Ammann, and Jeff Offutt. Adding Criteria-Based Tests to TDD. *7th Testing: Academic & Industrial Conference: Practice & Research Techniques (TAIC PART 2012)*, April 2012, Montreal, Canada.
94. **Nan Li**, Fei Li, and Jeff Offutt. Better Algorithms to Minimize the Cost of Test Paths. *5th IEEE International Conference on Software Testing, Verification and Validation (ICST 2012)*, April 2012, Montreal, Canada.

95. **Gary Kaminski**, Paul Ammann, and Jeff Offutt. Better Predicate Testing. *Sixth Workshop on Automation of Software Test (AST 2011)*, pages 57-63, May 2011, Honolulu, Hawaii, USA.
96. Jingyu Hu, **Nan Li**, and Jeff Offutt. An Analysis of OO Mutation Operators. *Seventh Workshop on Mutation Analysis (Mutation 2011)*, March 2011, Berlin, Germany.
97. Jeff Offutt, **Nan Li**, Paul Ammann, and **Wuzhi Xu**. Using Abstraction and Web Applications to Teach Criteria-Based Test Design. *24th IEEE-CS Conference on Software Engineering Education and Training*, May 2011, Honolulu, Hawaii, USA.
98. **Supaporn Kansomkeat**, Phachayanee Thiket, and Jeff Offutt. Generating Test Cases from UML Activity Diagrams using the Condition-Classification Tree Method. *2nd International Conference on Software Technology and Engineering*, October 2010, San Juan, Puerto Rico, USA.
99. **Garrett Kaminski**, **Upsorn Praphamontripong**, Paul Ammann, Jeff Offutt. An Evaluation of the Minimal-MUMCUT Logic Criterion and Prime Path Coverage. The 2010 International Conference on Software Engineering Research and Practice, Las Vegas Nevada, USA.
100. **Upsorn Praphamontripong** and Jeff Offutt. Applying Mutation Testing to Web Applications. *Sixth Workshop on Mutation Analysis (Mutation 2010)*, April 2010, Paris, France.
101. Pedro Reales Mateo, Macario Polo Usaola, and Jeff Offutt. Mutation at System and Functional Levels. *Sixth Workshop on Mutation Analysis (Mutation 2010)*, April 2010, Paris, France.
102. **Nan Li**, **Upsorn Praphamontripong**, and Jeff Offutt. An Experimental Comparison of Four Unit Test Criteria: Mutation, Edge-Pair, All-uses and Prime Path Coverage. *Fifth Workshop on Mutation Analysis (Mutation 2009)*, April 2009, Denver CO.
103. **Shuang Wang** and Jeff Offutt. Comparison of Unit-Level Automated Test Generation Tools. *Fifth Workshop on Mutation Analysis (Mutation 2009)*, April 2009, Denver CO.
104. **Supaporn Kansomkeat**, Jeff Offutt, **Aynur Abdurazik**, and Andrea Baldini. A Comparative Evaluation of Tests Generated from Different UML Diagrams. *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 867-872, August 2008, Phuket Thailand.
105. **Birgitta Lindström**, Jeff Offutt, and Sten Andler. Testability of Dynamic Real-Time Systems: An Empirical Study of Constrained Execution Environment Implications. *1st IEEE International Conference on Software Testing, Verification and Validation (ICST 2008)*, pages 112-120, April 2008, Lillehammer, Norway.
106. Jeff Offutt, Qingxiang Wang, and Joann J. Ordille. An Industrial Case Study of Bypass Testing on Web Applications. *1st IEEE International Conference on Software Testing, Verification and Validation (ICST 2008 Industry Track)*, pages 465-474, April 2008, Lillehammer, Norway.
107. **Birgitta Lindström**, Paul Pettersson, and Jeff Offutt. Generating Trace-Sets for Model-based Testing. *18th IEEE International Symposium on Software Reliability Engineering (ISSRE 2007)*, pages 171-180, November 2007, Trollhattan, Sweden.
108. Mohammad Abu-Matar and Jeff Offutt. Service Oriented Architecture Empirical Study. *The Nineteenth International Conference on Software Engineering and Knowledge Engineering (SEKE 2007)*, pages 628-631, July 2007, Boston MA, USA.
109. **Robert Nilsson** and Jeff Offutt. Automated Testing of Timeliness : A Case Study. *Second Workshop on Automation of Software Test (AST 2007)*, May 2007, Minneapolis, MN, USA.
110. **Mats Grindal**, Jeff Offutt, and Jonas Mellin. Managing Conflicts When Using Combination Strategies to Test Software. Australian Software Engineering Conference ASWEC 2007, pages 255-264, April 2007, Melbourne, Australia.
111. **Mats Grindal** and Jeff Offutt. Input Parameter Modeling for Combination Strategies. *IASTED International Conference on Software Engineering (SE 2007)*, February 2007, Innsbruck, Austria.
112. Jeff Offutt, Paul Ammann, and Lisa (Ling) Liu. Mutation Testing implements Grammar-Based Testing. *Second Workshop on Mutation Analysis (Mutation 2006)*, pages 93-102, November 2006, Raleigh, NC.
113. **Jing Guan**, Jeff Offutt, and Paul Ammann. An Industrial Case Study of Structural Testing Applied to Safety-critical Embedded Software. *ACM International Symposium on Empirical Software Engineering, ISESE 2006*, pages 272-277, September 2006, Rio de Janeiro, Brazil.

114. **Mats Grindal**, Jeff Offutt, and Jonas Mellin. On the Testing Maturity of Software Producing Organizations. *Testing: Academia & Industry Conference - Practice And Research Techniques, TAIC PART 2006*, pages 171-180, August 2006 Windsor, UK.
115. Jeff Offutt, **Yu-Seung Ma**, and Yong-Rae Kwon. The Class-Level Mutants of MuJava. *Workshop on Automation of Software Test (AST 2006)*, pages 78-84, May 2006, Shanghai, China.
116. Leonard Gallagher and Jeff Offutt. Automatically Testing Interacting Software Components. *Workshop on Automation of Software Test (AST 2006)*, pages 57-63, May 2006, Shanghai, China.
117. **Aynur Abdurazik** and Jeff Offutt. Coupling-based Class Integration and Test Order. *Workshop on Automation of Software Test (AST 2006)*, pages 50-56, May 2006, Shanghai, China.
118. **Robert Nilsson**, Jeff Offutt, and Jonas Mellin. Test Case Generation for Testing of Timeliness. *Second Workshop on Model Based Testing*. pages 102-121, March 2006, Vienna, Austria.
119. **Supaporn Kansomkeat**, Jeff Offutt, and Wanchai Rivepiboon. Class-Component Testability Analysis. *5th WSEAS International Conference on Software Engineering, Parallel & Distributed Systems*. February 2006, Madrid Spain.
120. **Wuzhi Xu**, Jeff Offutt, and Juan Luo. Testing Web Services by XML Perturbation. *IEEE International Symposium on Software Reliability Engineering*, pages 257-266, November 2005, Chicago Illinois.
121. **Supaporn Kansomkeat**, Jeff Offutt, and Wanchai Rivepiboon. Increasing Class-component Testability. *The IASTED International Conference on Software Engineering*, pages 156-161, February 2005, Innsbruck Austria.
122. Jeff Offutt, Ye Wu, Xiaochen Du, and **Hong Huang**. Bypass Testing of Web Applications. *IEEE International Symposium on Software Reliability Engineering*, pages 187-197, November 2004, Bretagne France.
123. **Robert Nilsson**, Jeff Offutt, and Sten F. Andler. Mutation-based Testing Criteria for Timeliness. *28th Annual International Computer Software and Applications Conference (COMPSAC 2004)*. September 2004, pages 306-311, Hong Kong.
124. Jeff Offutt and **Wuzhi Xu**. Generating Test Cases for Web Services Using Data Perturbation. *Workshop on Testing, Analysis and Verification of Web Services*. July 2004, Boston Mass.
125. Paul Ammann, Jeff Offutt, and **Hong Huang**. Coverage Criteria for Logical Expressions. *2003 International Symposium on Software Reliability Engineering (ISSRE '03)*. pages 99-107, Denver, CO, November 2003.
126. Ye Wu and Jeff Offutt. Maintaining Evolving Component-Based Software with UML. *Seventh European Conference on Software Maintenance and Reengineering (CSMR'03)*. pages 133-142, Benevento, Italy March 2003.
127. Ye Wu, Mei-Hwa Chen, and Jeff Offutt. UML-based Integration Testing for Component-based Software. *The 2nd International Conference on COTS-Based Software Systems (ICCBSS)* . pages 251-260, Ottawa, Canada, February 2003.
128. Mahmoud Elish and Jeff Offutt. The Adherence of Open Source Java Programmers to Standard Coding Practices. *The 6th IASTED International Conference Software Engineering and Applications*. Cambridge, MA, November 2002.
129. **Roger T. Alexander**, Jeff Offutt, and James M. Bieman. Syntactic Fault Patterns in OO Programs. *2002 International Conference on Engineering of Complex Computer Software*. pages 193-202, Greenbelt, MD, November 2002.
130. **Lisa Ferrett** and Jeff Offutt. An Empirical Comparison of Modularity of Procedural and Object-Oriented Software. *2002 International Conference on Engineering of Complex Computer Software*. pages 173-182, Greenbelt, MD, November 2002.
131. **Roger T. Alexander**, Jeff Offutt, and James M. Bieman. Fault Detection Capabilities of Coupling-based OO Testing. *2002 International Symposium on Software Reliability Engineering (ISSRE '02)*. pages 207-218, Annapolis, MD, November 2002.
132. **Yu-Seung Ma**, Yong-Rae Kwon, and Jeff Offutt. Inter-Class Mutation Operators for Java. *2002 International Symposium on Software Reliability Engineering*. pages 352-363, Annapolis, MD, November 2002.

133. Jeff Offutt. Web Software Applications Quality Attributes. *Quality Engineering in Software Technology (CONQUEST 2002)*, pages 187-198, Nuremberg, Germany, September 2002.
134. Stephen R. Schach, Bo Jin, David R. Wright, Gillian Z. Heller, and Jeff Offutt. Dependencies Within the Linux Kernel. *The ACM Mid-Southeast Chapter Fall Conference*, Gatlinburg, TN, November 2001.
135. Jeff Offutt, **Roger Alexander**, Ye Wu, Quansheng Xiao, and Chuck Hutchinson. A Fault Model for Subtype Inheritance and Polymorphism. *The Twelfth IEEE International Symposium on Software Reliability Engineering (ISSRE '01)*, pages 84–95, Hong Kong, PRC, November 2001.
136. **Suet Chun Lee** and Jeff Offutt. Generating Test Cases for XML-based Web Component Interactions Using Mutation Analysis. *The Twelfth IEEE International Symposium on Software Reliability Engineering (ISSRE '01)*, pages 200–209, Hong Kong, PRC, November 2001.
137. **Zhenyi Jin** and Jeff Offutt. Deriving Tests From Software Architectures. *The Twelfth IEEE International Symposium on Software Reliability Engineering (ISSRE '01)*, pages 308–313, Hong Kong, PRC, November 2001.
138. Peter J. Denning, Ravi Athale, Nada Dabbagh, Daniel Menascé, Jeff Offutt, Mark Pullen, Steve Ruth, and Ravi Sandhu. Designing an IT College. *Seventh World Conference on Computers in Education (WCCE 2001)*, Copenhagen, Denmark, June 2001.
139. **Roger T. Alexander** and Jeff Offutt. Criteria for Testing Polymorphic Relationships. *The Eleventh IEEE International Symposium on Software Reliability Engineering (ISSRE '00)*, pages 15–23, San Jose, CA, October 2000.
140. Jeff Offutt and **Roland Untch**. Mutation 2000: Uniting the Orthogonal *Mutation 2000: Mutation Testing in the Twentieth and the Twenty First Centuries*, pages 45–55, San Jose, CA, October 2000.
141. **Aynur Abdurazik** and Jeff Offutt. Using UML Collaboration Diagrams for Static Checking and Test Generation. *Third International Conference on the Unified Modeling Language (UML '00)*, pages 383-395, York, England, October 2000.
142. Jeff Offutt, **Aynur Abdurazik**, and **Roger T. Alexander**. An Analysis Tool for Coupling-based Integration Testing. *The Sixth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '00)*, pages 172–178, Tokyo, Japan, September 2000.
143. **Aynur Abdurazik**, Paul Ammann, Wei Ding, and Jeff Offutt. Evaluation of Three Specification-based Testing Criteria. *The Sixth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '00)*, pages 179–187, Tokyo, Japan, September 2000.
144. **Michelle Lee**, Jeff Offutt, and **Roger T. Alexander**. Algorithmic Analysis of the Impacts of Changes to Object-oriented Software. *34th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS USA '00)*, pages 61–70, Santa Barbara, CA, August 2000.
145. Jeff Offutt and **Aynur Abdurazik**. Generating Tests from UML Specifications. *Second International Conference on the Unified Modeling Language (UML '99)*, pages 416–429, Fort Collins, CO, October 1999.
146. **Jane Huffman Hayes** and Jeff Offutt. Increased Software Reliability Through Input Validation Analysis and Testing. *The Tenth IEEE International Symposium on Software Reliability Engineering (ISSRE '99)*, pages 199–209, Boca Raton, FL, November 1999.
147. Jeff Offutt, **Yiwei Xiong**, and Shaoying Liu. Criteria for Generating Specification-based Tests. *Fifth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS '99)*, pages 119-131, Las Vegas, NV, October 1999.
148. **Roger T. Alexander** and Jeff Offutt. Analysis Techniques for Testing Polymorphic Relationships. *30th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS USA '99)*, pages 104-114, Santa Barbara, CA, August 1999.
149. **Jane Huffman Hayes** and Jeff Offutt. Input Validation Testing: A Requirements-Driven, System Level, Early Lifecycle Technique. *11th International Conference on Software Engineering & its Applications*, Paris France, December 1998.
150. **Roland H. Untch**, Mary Jean Harrold, and Jeff Offutt. TUMS: Testing Using Mutant Schemata. *35th Annual ACM Southeast Conference*, pages 174-181, Murfreesboro, TN, April 1997.

151. Paul Ammann and Jeff Offutt. Maintaining Knowledge Currency in the 21st Century. *10th Conference on Software Engineering Education and Training*, pages 161–172, Virginia Beach, VA, April 1997.
152. Jeff Offutt and **Li Li**. Algorithmic Analysis of the Impact of Changes to Object-Oriented Software. *1996 International Conference on Software Maintenance*, pages 171–184, Monterey, CA, November 1996.
153. **Zhenyi Jin** and Jeff Offutt. Coupling-based Integration Testing. *Second IEEE International Conference on Engineering of Complex Computer Systems*, pages 10–17, Montreal, Canada, October 1996. (Outstanding Paper Award)
154. Jeff Offutt and **Jie Pan**. Detecting Equivalent Mutants and the Feasible Path Problem. *Eleventh Annual Conference on Computer Assurance (COMPASS 96)*, pages 224–236, Gaithersburg, Maryland, June 1996.
155. Jeff Offutt and **Jane Hayes**. A Semantic Model of Program Faults. *ISSTA 96, the International Symposium on Software Testing and Analysis*, workshop track, pages 195–200, San Diego, CA, January 1996.
156. Jeff Offutt and **Alisa Irvine**. Testing Object-Oriented Software Using the Category-Partition Method. *Seventeenth International Conference on Technology of Object-Oriented Languages and Systems (TOOLS USA '95)*, pages 293–304, Santa Barbara, CA, August 1995.
157. **Zhenyi Jin** and Jeff Offutt. Integration Testing Based on Software Couplings. *Tenth Annual Conference on Computer Assurance (COMPASS 95)*, pages 13–23, Gaithersburg, Maryland, June 1995.
158. Jeff Offutt, **Jie Pan**, and Jeffrey M. Voas. Procedures for Reducing the Size of Coverage-based Test Sets. *Twelfth International Conference on Testing Computer Software*, pages 111–123, Washington, DC, June 1995.
159. Jeff Offutt. Practical Mutation Testing. *Twelfth International Conference on Testing Computer Software*. pages 99–109, Washington, DC, June 1995.
160. Jeff Offutt. A Practical System for Mutation Testing: Help for the Common Programmer. *25th Annual International Test Conference (ITC 94)*, pages 824–830, Washington, DC, October 1994.
161. Paul Ammann and Jeff Offutt. Using Formal Methods To Derive Test Frames in Category-Partition Testing. *Ninth Annual Conference on Computer Assurance (COMPASS 94)*, pages 69–80, Gaithersburg, Maryland, June 1994.
162. Paul Ammann, Hassan Gomaa, Jeff Offutt, David Rine, and Bo Sanden. A Five Year Perspective on Software Engineering Graduate Programs at George Mason University. *7th SEI Conference on Software Engineering Education*, Springer-Verlag Lecture Notes in Computer Science Volume 750, Jorge L. Díaz-Herrera (Ed.), pages 473–488, San Antonio, Texas, January 1994.
163. **Roland Untch**, Mary Jean Harrold, and Jeff Offutt. Mutation Analysis Using Program Schemata. *1993 International Symposium on Software Testing and Analysis*, pages 139–148, Cambridge, Massachusetts, June 1993.
164. Jeff Offutt, Gregg Rothermel, and **Christian Zapf**. An Experimental Evaluation of Selective Mutation. *Fifteenth International Conference on Software Engineering*, pages 100–107, Baltimore, Maryland, May 1993.
165. D. P. Jacobs, Jeff Offutt, and S. V. Muddana. A Computer Algebra System for Nonassociative Identities. *Hadronic Mechanics and Nonpotential Interactions*, Proceedings of the Fifth International Conference, pages 185–195, Cedar Falls, Iowa, Myung, H.C. (Ed.), Nova Science Publishers, Inc., New York 1993.
166. Jeff Offutt and **Roland Untch**. Integrating Research, Reuse, and Integration into Software Engineering Courses. *1992 SEI Conference on Software Engineering Education*, Springer-Verlag Lecture Notes in Computer Science Volume 640, C. Sledge (Ed.), pages 90–98, San Diego, California, October 1992.
167. Jeff Offutt, R. Pargas, **S. V. Fichter**, and P. Khambekar. Mutation Testing of Software Using a MIMD Computer. *1992 International Conference on Parallel Processing*, pages II-257–266, Chicago, Illinois, August 1992.
168. Jeff Offutt and **S. D. Lee**. How Strong is Weak Mutation?. *Fourth Symposium on Software Testing, Analysis, and Verification*. pages 200–213, Victoria, British Columbia, October 1991.

169. Jeff Offutt. Using Mutation Analysis to Test Software. *Seventh International Conference on Testing Computer Software*, pages 65–77, San Francisco, California, June 1990.
 170. Jeff Offutt and **E. J. Seaman**. Using Symbolic Execution to Aid Automatic Test Data Generation. *Fifth Annual Conference on Computer Assurance*, pages 12–21, Gaithersburg, Maryland, June 1990.
 171. Jeff Offutt. An Integrated System for Automatically Generating Test Data. *First International Conference on Systems Integration*, pages 694–701, Morristown, New Jersey, April 1990.
 172. Jeff Offutt. The Coupling Effect: Fact or Fiction? *Third Software Testing, Analysis, Verification Symposium*, pages 131–140, Key West, Florida, December 1989.
 173. B. J. Choi, R. A. DeMillo, E. W. Krauser, R. J. Martin, A. P. Mathur, Jeff Offutt, H. Pan, and E. H. Spafford. The Mothra Tool Set. *22nd Hawaii International Conference on System Sciences*, pages 275–284, Kailua-Kona, Hawaii, January 1989.
 174. Rich DeMillo and Jeff Offutt. Experimental Results of Automatically Generated Adequate Test Sets. *Sixth Annual Pacific Northwest Software Quality Conference*, pages 209–232, Portland, OR, September 1988.
 175. Rich DeMillo, Dany Guindi, Kim King, Mike M. McCracken, and Jeff Offutt. An Extended Overview of the Mothra Software Testing Environment. *Second Workshop on Software Testing, Verification, and Analysis*, pages 142–151, Banff, Canada, July 1988.
 176. Cathy Bullard, Inez Caldwell, Jay Harrell, Cis Hinkle, and Jeff Offutt. Anatomy of a Software Engineering Project. *1988 SIGCSE Technical Symposium*, pages 129–133, Atlanta, GA, February 1988.
 177. Jeff Offutt and Kim King. A Fortran 77 Interpreter for Mutation Analysis. *1987 ACM SIGPLAN Symposium on Interpreters and Interpretive Techniques*, pages 177–188, St. Paul, MN, June 1987.
- REFEREED SHORT WORKSHOP PUBLICATIONS
(My students' names are *highlighted*.)
178. **Yu-Seung Ma**, Jeff Offutt, and Yong-Rae Kwon. MuJava: A Mutation System for Java. 28th International Conference on Software Engineering (ICSE '06), tool demo, pages 827-830. May 2006, Shanghai, China. (4 pages)
 179. Jeff Offutt, Ye Wu, **Xiaochen Du**, and **Hong Huang**. Web Application Bypass Testing. *First International Workshop on Quality Assurance and Testing of Web-Based Applications*. September, 2004, Hong Kong. (5 pages)
 180. Jeff Offutt, **Yu-Seung Ma**, and Yong-Rae Kwon. An Experimental Mutation System for Java. *Workshop on Empirical Research in Software Testing*. Boston, MA, July 2004. (4 pages)
 181. Jeff Offutt, **Yuan Yang**, and **Jane Huffman Hayes**. SEWeb: Making Experimental Artifacts Available. *Workshop on Empirical Research in Software Testing*. Boston, MA, July 2004. (4 pages)
 182. **Birgitta Lindström**, **Mats Grindal**, and Jeff Offutt. Using an Existing Suite of Test Objects: Experience from a Testing Experiment. *Workshop on Empirical Research in Software Testing*. Boston, MA, July 2004. (4 pages)
 183. **Jane Huffman Hayes** and Jeff Offutt. Applying a Semantic Fault Model to the Empirical Study of Corrective Maintenance. *Eighth IEEE Workshop on Empirical Studies of Software Maintenance*. pages 15-20, Montreal, Canada, October 2002. (4 pages)
 184. Stephen R. Schach and Jeff Offutt. On the Nonmaintainability of Open-Source Software. *The 2nd Workshop on Open-Source Software Engineering (OSSE 2002)*, <http://opensource.ucc.ie/icse2002/>, Orlando, FL, May 2002.
 185. **Jane H. Hayes** and Jeff Offutt. Product and Process: Key Areas Worthy of Software Maintainability Empirical Study. *International Workshop on Empirical Studies of Software Maintenance (WESS '00)*, http://members.aol.com/_ht_a/geshome/wess2000/metricsandmodels.htm, San Jose, CA, October 2000.
 186. **Aynur Abdurazik**, **Zhenyi Jin**, Liz White, and Jeff Offutt. Analyzing Software Architecture Descriptions to Generate System-level Tests. *Workshop on Evaluating Software Architectural Solutions (WESAS '00)*, <http://www.isr.uci.edu/events/wesas2000/>, Irvine, CA, May 2000.

187. Jeff Offutt. Software Testing: From Theory to Practice. IEEE AES Systems Magazine, June 1998. (3 pages. Based on a panel position paper at COMPSAC 1997: *Software testing: State-of-the-art Vs. State-of-the-practice.*)

- OTHER PUBLICATIONS – UNREFEREED PAPERS

188. **Jeff Offutt**. Regular editorials in Wiley's journal of Software Testing, Verification, and Reliability. March 2007-present. Available on my website (<http://cs.gmu.edu/offutt/stvr/>).
189. Jeff Offutt. Comments on Tai. *ACM Software Engineering Notes*, January 1990.
190. Jeff Offutt. Hints on Writing Style for Usenet. Monthly electronic posting to the Usenet newsgroup *news.announce.newusers*.
191. Jeff Offutt and C. Funsch. Lab Manual 1.0 for Computer Science 241: Data Structures. Clemson University, August 1990.

- TECHNICAL REPORTS

(My students' names are **highlighted**.)

192. **Nan Li** and Jeff Offutt, A Test Automation Language for Behavioral Models October 2013, GMU-CS-TR-2013-7, <http://cs.gmu.edu/~tr-admin/>.
193. Leonard Gallagher and Jeff Offutt, Integration Testing of Object-oriented Components Using FSMS: Theory and Experimental Details, July 2004, ISE-TR-04-04, <http://cs.gmu.edu/~tr-admin/>.
194. **Aynur Abdurazik**, Jeff Offutt, and Andrea Baldini, A Controlled Experimental Evaluation of Test Cases Generated from UML Diagrams, May 2004, ISE-TR-04-03, <http://cs.gmu.edu/~tr-admin/>.
195. Ye Wu, Jeff Offutt, and **Xiaochen Du**, Modeling and Testing Dynamic Aspects of Web-based Applications, March 2004, ISE-TR-04-01, <http://cs.gmu.edu/~tr-admin/>.
196. Ye Wu and Jeff Offutt, Modeling and Testing Web-based Applications, November 2002, ISE-TR-02-08, <http://cs.gmu.edu/~tr-admin/>.
197. Jeff Offutt, Generating Test Data From Requirements/Specifications: Phase IV Final Report, July 2001, George Mason University Department of ISE Technical Report ISE-TR-01-03, <http://cs.gmu.edu/~tr-admin/>.
198. Jeff Offutt, Generating Test Data From Requirements/Specifications: Phase III Final Report, May 2000, George Mason University Department of ISE Technical Report ISE-TR-00-02, <http://cs.gmu.edu/~tr-admin/>.
199. **Aynur Abdurazik** and Jeff Offutt, Generating Test Cases from UML Specifications November 1999, ISE-TR-99-09, <http://cs.gmu.edu/~tr-admin/>.
200. Jeff Offutt, Generating Test Data From Requirements/Specifications: Phase II Final Report, January 1999, George Mason University Department of ISSE Technical Report ISSE-TR-99-01, <http://cs.gmu.edu/~tr-admin/>.
201. Jeff Offutt, Generating Test Data From Requirements/Specifications: Phase I Final Report, April 1998, George Mason University Department of ISSE Technical Report ISSE-TR-98-01, <http://cs.gmu.edu/~tr-admin/>.
202. Jeff Offutt, Jeff Payne, and Jeffrey M. Voas, Mutation Operators for Ada, March 1996, George Mason University Department of ISSE Technical Report ISSE-TR-96-09.
203. **Li Li** and Jeff Offutt, Applying Logic-based Database to Impact Analysis of Object-oriented Software, September 1996, George Mason University ISSE Department Technical Report ISSE-TR-96-08.
204. Jeff Offutt and Jeffrey M. Voas, Subsumption of Condition Coverage Techniques by Mutation Testing, January 1996, George Mason University ISSE Department Technical Report ISSE-TR-96-01.
205. **Zhenyi Jin** and Jeff Offutt, Integrating Testing with the Software Development Process, August 1995, George Mason University ISSE Department Technical Report ISSE-TR-95-112.

206. Jeff Offutt, **Zhenyi Jin**, and **Jie Pan**, The Dynamic-domain Reduction Approach to Test Data Generation: Design and Algorithms, September 1994, George Mason University Department of ISSE Technical Report ISSE-TR-94-110.
207. Jeff Offutt and **Kanupriya Tewary**, Empirical Comparisons of Data Flow and Mutation Testing, April 1993, George Mason University Department of ISSE Technical Report ISSE-TR-93-101.
208. Paul Ammann and Jeff Offutt, Functional and Test Specifications for the MiStix File System, January 1993, George Mason University Department of ISSE Technical Report ISSE-TR-93-100.
209. Jeff Offutt and **David L. Pressley**. A Data Flow Oriented Approach to the Path Expression Constraint Generation Problem. Clemson University Computer Science Technical Report 92-113, January 1992.
210. Jeff Offutt and **Scott V. Fichter**. A Parallel Interpreter for the Mothra Mutation Testing System. Clemson University Computer Science Technical Report 92-100, January 1992.
211. Jeff Offutt and **S. D. Lee**. IMSCU Programmer's Reference Manual. Clemson University Computer Science Technical Report 91-121, July 1991.
212. D. P. Jacobs, Jeff Offutt, S. V. Muddana, and K. Prabhu. Version 1.0 Albert User's Guide. Technical Report 91-113, Department of Computer Science, Clemson University, March 1991.
213. Jeff Offutt and **C. Funsch**. Lab Manual 1.0 for Computer Science 241: Data Structures. Clemson University, August 1990.
214. R. A. DeMillo, D. S. Guindi, K. N. King, E. W. Krauser, W. M. McCracken, Jeff Offutt, and E. H. Spafford. Mothra Internal Documentation, Version 1.5. Technical Report SERC-TR-89/01, Software Engineering Research Center, Purdue University, 1989.
215. R. A. DeMillo, D. S. Guindi, K. N. King, E. W. Krauser, R. J. Martin, W. M. McCracken, Jeff Offutt, and E. H. Spafford. The Mothra Testing Environment User's Manual. Technical Report SERC-TR-4-P, Software Engineering Research Center, Purdue University.
216. Jeff Offutt, J. C. Flaspohler, and R. M. Harder. The Software Test and Evaluation Project Tools Baseline. Technical Report GIT-ICS-85/08, School of Information and Computer Science, Georgia Institute of Technology, 1985.

TEACHING EXPERIENCE

George Mason University Teaching Excellence Award, Teaching With Technology, 2013. Outstanding Teaching Award, School of Information Technology and Engineering, George Mason University, 2003.

Thirty-two years of teaching experience at three universities: George Mason University (GMU), Clemson University (CU), and the Georgia Institute of Technology (GIT). Taught one undergraduate course at GIT, a mix of graduate and graduate courses at CU and mostly graduate courses at GMU. Most courses at both schools were in the Software Engineering area.

Introduced a large number of innovations in terms of topics, teaching techniques, and style. This includes developing six completely new courses, substantially revising six others, and publishing four education related papers. Also pioneered several innovations for incorporating the web into class material delivery, some of which are now used by many other faculty. My course materials SWE 205, SWE 432, SWE 619, SWE 632, SWE 637, and SWE 642 have been used at Mason and universities around the world.

- Supervised Doctoral Students:
 - Lin Deng. *Mutation Testing for Android Applications*, August 2017. Assistant Professor, Towson University.
 - Upsorn Praphamontripong. *Testing Web Applications with Mutation Analysis*, May 2017. Assistant Professor, University of Virginia.
 - Jing Guan, *A Model-Based Testing Technique For Component-Based Real-Time Embedded Systems*, May 2015. Senior Software Engineer, Lockheed Martin.
 - Nan Li, *Generating Cost-Effective Criteria-Based Tests From Behavioral Models*, June 2014. Senior Software Engineer, Medidata Solutions.

- Jing Jin, *Towards Evasive Attacks: Anomaly Detection Resistance Analysis On The Internet*, December 2013. Senior Software Engineer – Security @ Intuit
 - Gary Kaminski, *Applications of Logic Coverage Criteria and Logic Mutation to Software Testing*, 2010. Primary advisor: Paul Ammann. Software Engineer at CACI.
 - Birgitta Lindström of Skövde University in Sweden. *Testability of Dynamic Real-Time Systems*. Graduated from Skövde University in Sweden, March 2009. (Co-advised with Dr. Sten Andler.) Associate Professor at Skövde University in Sweden.
 - Aynur Abdurazik, *Coupling Analysis of Object-oriented Software*, May 2007.
 - Recipient of ITEA Fellowship, 2002. Senior Software Engineer at NASA/SAIC.
 - Supaporn Kansomkeat, *An Analysis Technique to Increase Testability of Class-Component*. Graduated from Chulalongkorn University in Thailand, May 2007. (Co-advised with Dr. Wanchai Rivepiboon.) Associate Professor, Prince of Songkla University, Thailand.
 - Mats Grindal, *Evaluation of Combination Strategies for Practical Testing*. Graduated from Skövde University in Sweden, March 2007. (Co-advised with Dr. Sten Andler.) AddQ Consulting, Sweden.
 - Robert Nillson, *Mutation-Based Testing of Real-Time Software*. Graduated from Skövde University in Sweden, October 2006. (Co-advised with Dr. Sten Andler.) Google Switzerland.
 - Yu-Seung Ma, *Inter-Class Testing Using Mutation*. Graduated from KAIST University in Korea, 2005. (Co-advised with Dr. Yong-Rae Kwon.)
 - Roger Alexander, *Testing the Compositional Relationships of Object-oriented Components*, May 2001. Lead Software Engineer at Schweitzer Engineering Labs.
 - Zhenyi Jin, *Software Architecture-based Testing*, November 2000. ITT Industries.
 - GMU CS Distinguished Master's Graduate, 1994
 - Recipient of ITEA Fellowship, 1995. Systems Engineering Manager at Harris Corporation.
 - Michelle Lee (Li Li), *Object-oriented Change Impact Analysis*. November 1998. Executive Director, Web & Mobile Technology/Business Technology at The College Board.
 - Jane Hayes, *Input Validation Testing: A System Level, Early Lifecycle Technique*. September 1998. Professor, University of Kentucky.
 - Roland Untch, *Schema-based Mutation Analysis*, December, 1995. (Co-advised with Dr. M. J. Harrold). Professor, Middle Tennessee State University.
- Supervisor, Current Doctoral Students:
 - Kesina Baral
 - Anders Eriksson (Skövde University) *Model-Based Testing of Aeronautics Software* (Expected 2019)
 - Yun Guo. *Localizing and Fixing Faults in SQL Predicates* (Expected 2018)
 - Curriculum Development
 - Participated in a major restructuring of the **PhD in Computer Science**, effective Fall 2018 (rewrote first draft of document)
 - Participated in a major restructuring of the **MS in Software Engineering**, effective Fall 2018
 - Co-led team to create an **undergraduate concentration** in Software Engineering, as part of the Applied Computing Science degree (approved Fall 2009)
 - Led team to create an **undergraduate minor** in Software Engineering (approved Fall 2006)
 - Led team to a major restructure of GMU's **MS program** in Software Engineering (approved Fall 2005)
 - Developed a graduate Certificate in Web Software Engineering
 - Led team to develop a **PhD Concentration** in Software Engineering within GMU's PhD in Information Technology
 - With Henry Hamburger of GMU's CS Department, led team to design GMU's **PhD in Computer Science**, effective Fall 2000

- Participated in major restructuring of Clemson University’s MS in Computer Science, eliminating requirement for MS thesis (wrote first draft of document)
- Participated in major restructuring of Clemson University’s PhD in Computer Science (wrote first draft of document)
- GMU courses
 - Developed new PhD course: Advanced Software Testing (SWE 737)
 - Developed new BS course: Software Engineering Usability Analysis and Design (SWE 205)
 - Developed new MS course: Software Engineering for the World Wide Web (SWE 642)
 - Developed new MS course: User Interface Design and Development (SWE 632)
 - Developing new BS course: Design and Implementation of Software for the Web (SWE 432, for undergraduate engineering majors)
 - Developing new BS course: Software Testing and Maintenance (SWE 437, for undergraduate engineering majors)
 - Developed new PhD course: Analysis of Software for Testing (IT 824)
 - Developed new PhD course: Experimental Software Engineering (SWE 763)
 - Developed new PhD course: Special Topics in Web-based Software (IT 825)
 - Data Structures (graduate background, substantially revised using Java and electronic program submission)
 - Software Testing (SWE 637, substantially revised)
 - Software Construction with Ada (SWE 619, substantially revised)

PROFESSIONAL ACTIVITIES

- Conference Organizing Committees
 - Co-founder and Founding Chair of Steering Committee IEEE International Conference on Software Testing, Verification and Reliability (ICST)
 - Program Chair, ICST 2009 (IEEE)
 - Program Chair, Mutation 2007 (IEEE)
 - PhD Symposium Chair, ICST 2013, ISSRE 2015, ICST 2017
 - Chair, Ninth IEEE Workshop on Empirical Studies of Software Maintenance
 - Program Chair, 2001 International Conference on Engineering of Complex Computer Software
 - Treasurer, 1996 ACM/IEEE Computer Assurance Conference
- Journal Editorial Boards
 - Editor-in-Chief, The Journal of Software Testing, Verification, and Reliability (since 2006)
 - IEEE Transactions on Software Engineering (2001-2005)
 - The Journal of Software and Systems Modeling (since 2004)
 - Empirical Software Engineering Journal (since 2006)
 - Software Quality Journal (since 2002)
- Selected Conference Technical Program Committees
 - Mutation 2013, COUFLESS 2013
 - ICST 2013, Mutation 2013, ICWS 2013
 - ICST 2013, ISSRE 2013, Mutation 2013, ICWS 2013, MODELSWARD 2013
 - ISSRE 2012, A-MOST 2012, Mutation 2012, WSOFL 2012, TAP 2012
 - ICST 2011, TAICPart 2011, Mutation 2011, A-Most 2011

- 3rd International Conference on Software Testing, Verification and Validation (ICST 2010)
- Mutation Workshop 2010
- Workshop on Test Driven Development (co-Program Chair, TDD 2010)
- Workshop on 6th Advances in Model-Based Testing (A-MOST 2010)
- Testing: Academic and Industrial Conference - Practice and Research Techniques (TAIC PART 2010)
- 4th workshop on Testing, Analysis, and Verification of Web Software (TAV-WEB 2010)
- 2nd International Conference on Software Testing, Verification and Validation (Program Chair, ICST 2009)
- Mutation Workshop 2009
- Conference On Quality Software (QSIC 2009)
- 18th International Symposium on Software Reliability Engineering (ISSRE 2009)
- Workshop on Model-Driven Engineering, Verification, and Validation (MODEVVA 2009)
- 1st International Conference on Software Testing, Verification and Validation (ICST 2008)
- 16th International Symposium on Software Reliability Engineering (ISSRE 2007)
- Automated Software Testing (AST 2007)
- Second International Workshop on Random Testing (RT 2007)
- International Workshop on Software Test Evaluation (STEV 2007)
- Conference On Quality Software (QSIC 2007)
- Workshop on Advances in Model-Oriented System Testing (A-MOST 2007)
- Workshop on Model Based Testing (MBT 2007)
- Domain-Specific Approaches to Software Test Automation (DoStA 2007)
- Tests And Proofs (PAT 2007)
- 15th International Symposium on Software Reliability Engineering (ISSRE 2006)
- Formal Approaches to Testing of Software (FATES 2006)
- Workshop on Quality Assurance and Testing of Web-based Applications (QATWEB 2006)
- Workshop on Model Based Testing (MBT 2006)
- Automated Software Testing (AST 2006)
- IEEE International Conference on Web Services (ICWS 2006)
- The Role of Software Architecture for Testing and Analysis (ROSATEA 2006)
- 16th International Symposium on Software Reliability Engineering (ISSRE 2005)
- Formal Approaches to Testing of Software (FATES 2005)
- Workshop on Quality Assurance and Testing of Web-based Applications (QATWEB 2005)
- Conference On Quality Software (QSIC 2006)
- IEEE International Conference on Software, Science, Technology and Engineering (SwSTE 2005)
- 15th International Symposium on Software Reliability Engineering (ISSRE 2004)
- 6th International Conference on Formal Engineering Methods (ICFEM 2004)
- Formal Approaches to Testing of Software (FATES 2004)
- 28th Annual International Computer Software and Application Conference (COMPSAC 2004)
- Workshop on Quality Assurance and Testing of Web-based Applications (QATWEB 2004)
- Testing, Analysis, and Verification of Web Services (TAV-WEB 2004)
- Workshop on Model Based Testing (MBT 2004)
- 15th IFIP International Conference on Testing of Communicating Systems (TestCom 2004)
- 14th International Symposium on Software Reliability Engineering (ISSRE 2003)

- 5th International Conference on Formal Engineering Methods (ICFEM 2003)
 - 10th Asia Pacific Software Engineering Conference (APSEC 2003)
 - 1st Workshop on Verification and Validation of Enterprise Information Systems (2003)
 - 13th International Symposium on Software Reliability Engineering (ISSRE 2002)
 - 2002 International Conference on Engineering of Complex Computer Software (ICECCS 2002)
 - 4th International Conference on Formal Engineering Methods (ICFEM 2002)
 - Formal Approaches to Testing of Software (FATES 2002)
 - 9th Asia Pacific Software Engineering Conference (APSEC 2002)
 - Doctoral Symposium Subcommittee for ICSE 2001
 - 2001 International Conference on Engineering of Complex Computer Software (ICECCS 2001)
 - 8th Asia Pacific Software Engineering Conference
 - Mutation 2000: Mutation in the Twentieth and the Twenty First Century
 - 7th Asia Pacific Software Engineering Conference
 - 2000 International Conference on Engineering of Complex Computer Software
 - ICFEM 2000: International Conference on Formal Engineering Methods
 - 2000 IEEE/SEI Conference on Software Engineering Education & Training
 - 2000 International Conference on Testing Computer Software
 - 1999 Working Conference on Reverse Engineering
 - 1999 International Conference on Engineering of Complex Computer Software
 - 1999 International Conference on Software Maintenance
 - 1999 IEEE/SEI Conference on Software Engineering Education & Training
 - 1998 IEEE/SEI Conference on Software Engineering Education & Training
 - 1997 IEEE International Conference on Formal Engineering Methods
 - 1997 International Conference Engineering Complex Computer Systems
 - 1997 International Conference on Software Maintenance
 - 1996 ACM International Symposium on Software Testing and Analysis
 - 1990 and 1991 ACM Southeast Regional Conference
- Proposal Refereeing
 - Panelist for NSF Software Engineering and Languages program, 2007
 - Panelist for NSF SBIR Software Engineering program, 2001
 - Panelist for NSF Information Technology Research, 2000
 - Panelist for NSF Software Engineering program, 1996
 - Panelist for NSF SBIR program, 1993
- Journal Refereeing
 - IBM Systems Journal
 - IEEE Transactions on Computers
 - IEEE Transactions on Software Engineering
 - ACM Transactions on Software Engineering Methodology
 - IEEE Software
 - Journal of Software Testing, Verification, and Reliability
 - Software Practice and Experience
 - The Journal of Systems and Software
 - Journal of Systems Integration

- Journal of Information and Systems Engineering
- Conference Refereeing (not on technical program committee)
 - ACM Southeast Regional Conference, 2000
 - International Symposium on Software Reliability Engineering, 1995
 - SEI Conference on Software Engineering Education, 1994, 1995
 - Workshop on Experience with Building Distributed and Multiprocessor Systems, 1989
 - International Conference on Systems Integration, 1990
 - International Conference on Testing Computer Software, 1990
 - ACM Computer Science Conference, 1988, 1991
 - Third International Conference on Software Engineering and Knowledge Engineering, 1991
 - International Conference on Parallel Programming, 1992
 - 15th International Conference on Software Engineering
- Professional Societies
 - Association for Computing Machinery
 - IEEE Computer Society

ACADEMIC SOFTWARE PROJECT EXPERIENCE

Throughout my career I have focused on software engineering techniques that incorporate a large theoretical component, and also that can be used by real programmers. I emphasize empirical validations, and as such, have constructed, helped to construct, or led the construction of several working software systems. Most of these offer a proof-of-concept demonstration vehicle of theoretical concepts as well as a lab for empirical validation. Some of these (most notably muJava, Mothra and Godzilla, and also Mistix and Albert) have been used by hundreds of other researchers. The Mothra project in particular was used by many dozens of researchers over a period of fourteen years.

- *Mutation Testing for Java (μ Java)*. This project is a testing tool that is distributed under an open source software model. μ Java is a mutation testing system for Java programs that supports the object-oriented features of inheritance, polymorphism and dynamic binding. It was built as an international collaborative effort between myself at GMU and Yu-Seung Ma at the Korean Advanced Institute for Science and Technology (KAIST), as part of her PhD work. μ Java is freely available from GMU (<https://cs.gmu.edu/~offutt/mujava/>) and as open source. It is currently being used at dozens of universities for research and teaching purposes. Three papers have been published about μ Java, it was featured at the research tool demo session at the International Conference on Software Engineering in May 2006, and at least two dozen papers by other researchers used μ Java.
- *Software Engineering Experimentation on the Web (SEWeb)*. This NSF-funded research project is creating a one-stop web site for downloading software engineering experimental artifacts. The projects will contain source code, requirements, design models, test cases, buggy versions, and other artifacts. An initial prototype is being built in Summer 2003 as part of an NSF Research Experience for Undergraduates project.
- *JOSCAT*. This tool measures coupling for object-oriented software.
- *Coverage-based Analysis Tool (CBAT)*. This NSF-funded research project created a fully functional, multi-capability, robust analysis and coverage tool for Java. This tool allows researchers to generate a large number of coverage graphs that represent Java programs, which can then be used for test generation, test coverage measurement, metrics computation, maintenance computations, and other purposes. CBAT focuses on inheritance and polymorphic relationships, as well as traditional control and data flow information. This tool was built as part of Roger Alexander's PhD dissertation.
- *Coupling-based Testing (CoupTest)*. This NSF-funded research project created an analysis tool to extract coupling relationships and measure coverage of tests according to the coupling-based test technique. This tool was built in Java for Java programs.

- *Specification-based Testing (SpecTest)*. This Rockwell/Collins-funded research project developed a tool to measure the extent to which system-level test data satisfies a set of test criteria that are defined on formal specifications and design models of the software. The tool works with the NRL's SCRTool, which implements the SCR specification language, and Rational Corporation's Rose tool, which supports the Unified Modeling Language. This tool was primarily built by Aynur Abdurazik as part of her MS thesis.
- *Program Mutation for SDI Applications (Mothra)*. This research project involved approximately 12 people in the design and development of a mutation-based unit-level software testing system known as Mothra. Mothra is over 50,000 lines of C code, executes in Unix environments, and among other tools includes a parser, an interpreter, a test harness, a test case manager, and several user interfaces. While a graduate student, Dr. Offutt and Dr. Rich DeMillo (currently of Purdue University) were the major designers of Mothra, and Offutt was responsible for over 50% of the implementation, including the interpreter, the test harness, the test case manager, and the initial user interfaces. This work has yielded numerous papers in refereed conferences and journals; Mothra has been used as a research and demonstration vehicle in over a dozen sites.
- *Godzilla*. This tool was designed and built as part of Offutt's doctoral work, Godzilla automatically generated test data for unit testing according to several test criteria, including mutation, multiple condition coverage, branch and statement. Godzilla was approximately 25,000 lines of C code, executed in Unix environments, and was fully integrated with Mothra. A software company in California, Agitar, has incorporated much of the Godzilla's test data generation innovations into their commercial testing tool, Agitator.
- *HyperMothra, Leonardo, and Equalizer*. These proof-of-concept systems all involve major modifications to Mothra to improve the efficiency of the mutation testing process. These systems provide extended functionality, use different basic algorithms, and utilize parallel hardware architectures. These projects were all implemented by graduate students as Master's projects, using Offutt's design and under his direction.
- *IMSCU*. This streamlined mutation system was built by several graduate students according to Offutt's specification and under his direction. Two versions have been built, one in C (about 8,000 lines of code) and another in Modula-2 (about 5,000 lines). IMSCU has been used in several course projects at both the graduate and undergraduate level, and as a research vehicle in software metrics experimentation. In an undergraduate senior-level project-oriented course, IMSCU was used as a *project template*, where the class was supplied with a partial implementation and a system architectural design, and divided into five teams, each of which derived requirements for, designed, implemented, and tested an additional major subsystem. All subsystems were integrated into a complete system, which was then tested as a whole. This project involved reuse, maintenance, integration, and all phases of the software life cycle, and was managed by Offutt to provide practical, industrial-oriented experience to the students.
- *Mistix*. This simplified file system was initially specified by Dr. Offutt for a class implementation project. It has been used in several classes for implementation projects, and has been implemented in several languages (C, Modula-2, Ada) by Offutt. Implementations have been used in test classes, user interface classes, and the concept has been used in formal methods classes. Implementations have also been used in several research projects.
- *Albert*. This project to build a computer algebra system for nonassociative identities was led by Dr. Jacobs of Clemson University. Offutt designed both the overall system and a command-language user interface, which were implemented by graduate students. This system is currently used by mathematicians around the world to support research in nonassociative algebra.

MISCELLANEOUS ACADEMIC ACTIVITIES

- External Curriculum Development
 - Invited as an external expert on Software Engineering Education to design an innovative undergraduate curriculum in Software Engineering for the United Arab Emirates University.

- Supervisor, Master's Theses:
 - Chandra Alluri, *Testing Calculation Engines Using Input Space Partitioning and Automation*, MS, Software Engineering, 2008.
 - Vasileios Papadimitriou, *Automating Bypass Testing for Web Applications*, MS, Software Engineering, 2006.
 - Aynur Abdurazik, *Specification-based Test Data Generation Using UML*, MS, Software Engineering, 1999.
 - Ammei Lee, *FGS: A Multi-purpose Laboratory for Software Engineer Research and Education*, MS, Computer Science, 1998.
 - Eleanor Rizzo, MA, Interdisciplinary Studies, 1998.
 - Alisa Irvine, *The Effectiveness of Category-partition Testing of Object-oriented Software*, 1994.
 - Jie Pan, *Using Constraints to Detect Equivalent Mutants*, 1994.
 - Christian Zapf, *Distributing Mutation on a Network of Sun Servers*, 1993.
 - Tracey Oakes, *A WIMP Interface to Mothra*, 1993.
 - David Pressley, *Data Flow Analysis for Generating Statement Coverage Constraints*, 1992.
 - Raad Yacu, *An Improved Procedure for Generating Statement Coverage Constraints*, 1991.
 - Scott Fichter, *Parallelizing Mutation on a Hypercube*, 1991.
 - Stephen D. Lee, *Weak vs. Strong: An Empirical Comparison of Mutation Variants*, 1991.
 - W. Michael Craft, *Detecting Equivalent Mutants Using Compiler Optimization Techniques*, 1989.
 - Jason Emil Seaman, *Using Symbolic Evaluation to Address the Internal Variable Problem*, 1989.
- George Mason University Committees
 - Member of Provost's Graduate Council, 2000-2006
 - Member of Faculty Senate Information Technology Committee, 2001-2002 – Provost's appointee
 - Advisor to the MA in Telecommunications, Information Systems track, 1994-1998
 - Provost's Task Force for Ethical Computing, Chair, 1994-1995
 - ⊙ Responsible for creating a new University Computing Use Policy document
- George Mason IT&E Service
 - Graduate Studies Committee 2009-2010
 - IT&E Tenure and Promotion Committee, 2005-2006
 - IT&E system administrator hiring committee, 2001
 - CS PhD Coordinating Committee, 1999-2002
 - Committee on Faculty Productivity, 1997
 - Dean Search Committee, 1997
 - IT&E Computing Environment Committee, 1994-present
 - Undergraduate Studies, 1993-1994
- George Mason CS Departmental Service
 - Software Engineering MS and PhD Coordinator, 2007-
 - Graduate Studies Committee 2007- (Chair 2009-2010)
 - Executive Committee 2007-
 - Space Committee 2009-
- George Mason ISE Departmental Service

- Software Engineering MS and PhD Coordinator, 2003-2007
- Hiring Committee 1997, 1998, 1999 (Chair), 2000 (Chair), 2006 (Chair)
- Acting Department Chair, 1998
- Assistant Department Chair, 1996
- ISE Webmaster, 1998-2000, 2004
 - Supervised graduate student
- Computing Infrastructure, 1993-2000
 - Created ISE computer use policies
 - Supervised ISE system administrator
- Undergraduate Studies (ISE committee), 1994-1998
- ISE Technical Report Editor, 1993-1994
 - Initiated the ISE technical report series
- B.S. in ISE, 1993-1994
- ISE Professional Courses, 1994-1995
- Academic Service (At Clemson)
 - Qualifying Exam (responsible for Operating Systems)
 - Lab Facilities
 - Graduate Affairs
 - Rewrote PhD program handbook, 1991
 - Redesigned MS program, 1990
 - Department Advisory
 - elected 1989, 1991
 - Department Head Search (elected)
- Graduate Student Advisory Committees
 - Margaret Francel (Georgia Tech, Phd CS, Outside Reader)
 - Wei Ding (George Mason, MS SWE)
 - Ron Durham (George Mason, MS CS)
 - Todd Baylor (George Mason, PhD)
 - Mark Blackburn (George Mason, PhD, Spring 1998)
 - Shawn Bohner (George Mason, PhD, Summer 1995)
 - Bill Brykczynski (George Mason, PhD, Spring 1999)
 - Pai Yen Chung (George Mason, PhD)
 - Ann Clessas (George Mason, PhD)
 - Joe Constantini (George Mason, PhD)
 - Chao Din (George Mason, PhD)
 - Ghulam Farrukh (George Mason, PhD, Spring 1998)
 - Pat Patterson (George Mason, PhD, Fall 1995)
 - William Pritchett (George Mason, PhD)
 - Indrakshi Mukherjee Ray (George Mason, PhD, Spring 1998)
 - Edwin Rueda (George Mason, PhD)
 - Michael Schoelles (George Mason, PhD)
 - Jeffrey Yang (George Mason, PhD)
 - Sheila Banks (Clemson, PhD, Spring 1995)
 - Veera Sekhar Muddana (Clemson, MS, Spring 1991)
 - Kirtikumar Prabhu (Clemson, MS, Summer 1991)
 - Ganesh Kadaba (Clemson, MS, Spring 1991)