CURRICULUM VITAE

Jeff Offutt, PhD

January 2025

Professor & Chair
Department of Computer Science
College of Nanotechnology, Science, & Engineering
University at Albany, Albany, NY 12222

offutt@albany.edu
www.linkedin.com/in/jeff-offutt/

h-index = 77; *i10-index* = 176 (Google Scholar, April 2025); 25,249 citations; *Erdös number* = 3

BIOGRAPHICAL SUMMARY

Jeff Offutt is Professor and Chair of the Department of Computer Science at University at Albany, with a PhD from the Georgia Institute of Technology and 37 years of experience as a professor. He conducts research in software engineering and CS education, and has contributed major results in software testing, model-based testing, and dynamic symbolic execution. He has more than 200 refereed publications and led large, multi-disciplinary, multi-institution, collaborative research projects with funding from government and industry. He has designed and taught courses and degree programs as all levels, succeeded in academic leadership roles both in universities and the profession; and won major awards that reflected research, teaching, and service. His research results are used widely in industry and by researchers, and his educational innovations are used around the world.

At the University at Albany, Offutt is managing unparalleled growth in the MS-CS program, handling continuing growth in the BS-CS program, and has engineered significant growth in the PhD-CS program. In the past two years, he has led efforts to modernize all three academic programs, hired five faculty members, hired new administrative staff, and restructured nearly all procedures and processes.

As Associate Chair of Graduate Studies at George Mason University, Offutt engineered growth of the MS programs from 150 students to 700 students in two years, and restructured the MS programs to successfully educate the large numbers of students.

Offutt has received several major awards for both research and teaching from university, state, and the research community. He has been awarded three Most Influential Papers and an ACM Notable paper. The most recent is a most influential paper of the second decade of IEEE Transactions on Software Engineering to be recognized in the 50th anniversary issue.

Dr. Offutt invents and develops techniques to develop high-quality software that is reliable, usable, ethical, and effective.

He has led a multi-disciplinary, multi-institution, NSF project to integrate Computer Science Standards of Learning into K-5 classrooms, as well as projects on automated software testing. His project on Testing of Critical System Characteristics (TOCSYC) involved scientists from three Swedish universities and two government agencies, and his Google-funded SPARC project created a new teaching model for CS1 and CS2 that increases diversity, scalability, and retention, while reducing plagiarism. He has led and worked on projects with funding from numerous government agencies and companies.

He has taught Software Engineering courses at all levels and has developed new courses on several Software Engineering subjects. His textbook, *Introduction to Software Testing* (with Paul Ammann), is the leading textbook in software testing.

Offutt has served on or led virtually every departmental academic committee. He co-chaired a university-wide committee on Graduate Education, which resulted in more than a dozen major recommendations for new graduate programs and structural changes throughout the university. He led the MS in Software Engineering program at GMU for 20 years, where it was rated as having the highest alumni satisfaction of any MS program across the university.

Offutt was editor-in-chief of Wiley's journal of Software Testing, Verification, and Reliability from 2006-2019 and co-founded the IEEE International Conference on Software Testing, Verification, and Validation (ICST). He has also served on dozens of conference program committees and is on the editorial boards for several research journals.

Offutt has given dozens of keynote and plenary talks, and has presented his research at universities and companies throughout the world. He has designed, built, and directed many software research systems, which have been used by thousands of software engineering researchers and educators. He has invented, developed, and experimentally validated numerous algorithms and engineering techniques in software testing, many of which are in widespread use.

PROFESSIONAL EXPERIENCE

- **Since August 2023**: Full Professor and Chair of the Computer Science Department, University at Albany
- **1992-2023**: Full Professor, Associate Chair of the Computer Science Department, and Program Director for the MS Software Engineering program, George Mason University. Associate Professor 1996–2005, Assistant Professor 1992–1996.
- **Part-time and Visiting Positions**:
  - **2002-2018**: Guest research professor in the Department of Computer Science, University of Skövde, Skövde Sweden
  - **2010-2013**: Guest research professor in the Department of Computer Science, University of Linköping, Linköping Sweden
  - **2000 to 2007**: Part-time Research Scientist with the National Institute of Standards and Technology's Information Technology Lab
- **1998 to 1999**: Chair of Information and Software Engineering Department, George Mason University
- **1988 to 1992**: Assistant Professor of Computer Science, Clemson University

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 (*major*), Data Processing (*major*), Physics (*minor*), Morehead State University, May 1982

FUNDED RESEARCH PROJECTS

- *Excellence in Computer Science Education and Learning (ExCSEL)*, SUNY Promoting Recruitment, Opportunity, Diversity, Inclusion, and Growth (PRODiG+) postdoc position grant. $160,000, Sept. 2024–Aug. 2026.
- *A Partnership to Implement the Inclusive Computer Science Model of Professional Development and a Digital Platform for PK-6 Computer Science Teaching for Students with Disabilities*, Co-PIs: Amy Hutchison, Anna Evmenova, and Erdogan Kaya, College of Education at GMU; Jamie Colwell, College of Education at Old Dominion University, NSF Computer Science for All program, $999,985, Oct. 2021–Sep. 2024
- *Preparing K-5 Teachers to Integrate the Computer Science Standards of Learning in Inclusive Classrooms to Support Students with High Incidence Disabilities*, Co-PIs: Amy Hutchison, Anna Evmenova, and Erdogan Kaya, College of Education at GMU; Jamie Colwell and Kristie Gutierrez, College of education at Old Dominion University; and Rebecca Dovi, CodeVA, NSF Computer Science for All program, $999,423, Oct. 2018–Sep. 2021
- *Testing of Critical System Characteristics (TOCSYC)*, Swedish Knowledge Foundation, 30,972,948 SEK (∼$4,700,000 USD), 2013-2018. A consortium of scientists from GMU, Mälardalen University, Skövde University, Blekinge Institute of Technology, Karlstad University, and SICS Swedish ICT AB. The project was led by Offutt, Paul Pettersson, and Sten Andler
- *SPARC: Self-PAced Learning increases Retention and Capacity*, Google Education Grant, 3X in 3 Years Awards program, $900,000, Feb. 2015–Feb. 2018

- *Usable Analysis of Security Protocols*, Department of Homeland Security, $128,993 (GMU's share) A multi-institution project with participants from GMU, University College London, Mitre Corporation, and Dartmouth College's I3P, Feb. 2012–Feb. 2014
- *Practical Test Automation*, Pyramid Inc, $20,000, May 2019–Aug. 2019. An industry-university cooperative project
- *PILOT*, Swedish Vinnova and Saab Aeronautics, 6,500,000 SEK (∼$1,000,000 USD), 2013–2015 (Collaborative with Skövde University)
- *Testing of Event-Triggered Real-Time Systems (TETReS)*, Swedish Knowledge Foundation. 8,500,000 SEK (∼$930,000 USD) (Collaborative with Skövde University), 2003–2008
- *Automating Input Space Partitioning to Test Calculation Engines*, Freddie Mac, $120,000, Aug. 2006–May 2008. An industry-university cooperative project
- *Research into Testing Service Oriented Architectures*, Avaya Research Labs, $61,169, Jul. 2006–May 2007
- *Coupling-based Object-Oriented Software Analysis*, NIST Information Technology Lab, $24,988, Jun. 2005–Jan. 2006
- *Coupling-based Analysis of Object-Oriented Software and Web Services*, NIST Information Technology Lab, $24,990, Mar. 2006–Aug. 2006
- *Assuring Web-based Software System Components*, NASA Goddard, $104,331, Sep. 2003–Dec. 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, Jun. 2002–Sep. 2002
  REU Supplemental Research Experience for Undergraduates: $6459, Jun. 2003–Sep. 2003
- *Coupling-based Analysis for Integration Testing of Object-oriented Software*, NSF: CCR–98 04111, $200,000, Jul. 1998–Jun. 2001
- *A Comparative Evaluation of Data Flow and Mutation Testing*, NSF: CCR–93 11967, $107,390, Aug. 1993–Jan. 1996
- *Assuring Web-based Software System Components*, NASA Software IV&V Facility: $136,000, 2001 (Note: Awarded but funding not released)
- *Generating Test Cases From Requirements/Specifications*, Rockwell-Collins Avionics, $125,000, May 1997–Aug. 2000
- *Estimator Efficiency*, Software Productivity Consortium, $29,200, Jan. 1999–May 1999
- *Software Reliability for Real-Time Control Systems*, NASA Langley Research Center, $20,000, Aug. 1989–Aug. 1990
- *A Practical Mutation Testing System for Ada*, NASA Phase I SBIR grant in collaboration with Reliable Software Technologies Corporation, Feb. 1993–Jul. 1993
- *Specification-based testing*, $500,000, The Ministry of Education of Japan under Joint Research Grant-in-Aid for International Scientific Research FM-ISEE (08044167) (with Dr. Shaoying Liu of Hosei University, Tokyo Japan), 1999-2001
- *XML Testing of Web Services*, IT&E Graduate Research Assistantship, Aug. 2005–May 2006
- *Testing Data State Interactions Web Software Applications*, IT&E Graduate Research Award, Aug. 2002–May 2003
- *Software Testing Based on Module Coupling*, IT&E Graduate Research Assistantship, Aug. 1995–May 1996

HONORS AND AWARDS

h-index = 77; *i10-index* = 176 (Google Scholar, April 2025); 25,249 citations; *Erdös number* = 3

- **Awards recognizing both research and teaching**
  - *Outstanding Faculty Award*, State Council of Higher Education for Virginia, 2019 (Virginia's highest award for university faculty)
  - *Faculty of the Year*, George Mason University Alumni Association, 2020
  - GMU *Outstanding Faculty member*, 2009, 2010
- **Research focused awards**
  - IEEE TSE's *Most Influential Paper* of the second decade, recognized in the 50-year anniversary issue (*Constraint-based Automatic Test Data Generation*, 1991)
  - ISSRE 30-year *Most Influential Paper* award (*Inter-class mutation operators for Java*, 2002)
  - SoSyM / MODELS *10-year Most Influential Paper* award (*Modeling presentation layers of web applications for testing*, 2010-2020)
  - ACM Computing Reviews *Notable Article* for 2013, *Putting the Engineering into Software Engineering Education*, IEEE Software 30(1), February 2013
  - *Best paper award*, International Conference on Software Testing, Verification, and Reliability, 2021
  - *Outstanding paper* award, Mutation Workshop 2018
  - *Outstanding Paper* award, ICECCS 1996
  - *Outstanding Researcher* award, School of IT&E, 2004
  - *Outstanding Researcher* award, ISE Department, 2006
- **Educator focused awards**
  - *John Toups Presidential Medal for Faculty Excellence in Teaching*, 2020
  - George Mason University *Teaching Excellence Award, Teaching With Technology*, 2013
  - Finalist, *Governor's Technology in Education* award, 2012
  - *Outstanding Teacher* award, ISE Department 2003, IT&E 2003

SIGNIFICANT INVENTIONS AND INNOVATIONS

Offutt and his collaborators have contributed numerous inventions and innovations in software engineering

- **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. Mutation is used by Meta, Google, Cisco Systems, SAAB Aerospace, JP-Morgan, among others, and incorporated into the chip testing tool Certitude.
  - With Drs. Paul Ammann and Marcio Delamaro, invented *minimal mutation*, with the potential to reduce the cost of mutation by one to two orders of magnitude
  - Developed two tools for widespread use; *Mothra* was used throughout the testing community in the 1990s and 1980s (my PhD work), *muJava* has been used by hundreds of researchers and students for 20 years (with former PhD 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 significant performance problems (with former PhD student Dr. Roland Untch)
  - Invented techniques to detect *equivalent mutants*, greatly reducing a bottleneck to the adoption of mutation (with former MS student Jie Pan)

  - Empirically demonstrated that *weak mutation* is a viable alternative to strong mutation (with former PhD student Stephen Lee)
  - Developed several processes for how best to apply mutation
  - Invented algorithms to parallelize mutation
  - Empirically showed that mutation is superior to several other testing techniques, including data flow, edge coverage, and logic coverage
  - Developed mutation operators for *real-time testing* (with former PhD 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 Microsoft's Pex.
  - Published the first algorithms for *symbolic execution* and co-invented (with Bogdan Korel) *dynamic symbolic execution*
  - 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*, still the strongest algorithm for generating test data

- **Model-based testing**: Invented the first testing techniques based on design models, including new criteria. This work helped spawn a new sub-field in software testing that has led to hundreds of papers, several conferences and workshops, and is used widely in the software industry (with former PhD student Dr. Aynur Abdurazik)
- **Bypass testing**: Invented an effective way to black-box test web applications for functional and security problems (bypass testing is part of the widely used commercial tool Selenium)
- **Input space partitioning**: Developed new methods and processes for applying input space partitioning (with former students Dr. Mats Grindal and Chandra Alluri of Freddie Mac)
- **Maintainability**: Developed empirical techniques for evaluating maintainability of software (with Dr. Steve Schach at Vanderbilt University)
- **Metrics**: Developed metrics for use with object-oriented software, software coupling, and component-based software
- **Web applications**: Developed several concepts for testing web applications and services, including bypass testing, data perturbation, finite state machine modeling, and atomic section modeling (with Dr. Ye Wu, Dr. Anneliese Andrews and former student Wuzhi Xu)
- **Prime path testing**: Invented a new way to design effective software tests based on graphs (with Dr. Paul Ammann)
- **Base choice testing**: Invented a new way to design effective software tests based on input spaces (with Dr. Paul Ammann)
- **Coupling-based testing for object-oriented software**: Invented new test criteria for integration testing of software
- **Object-oriented software testing**: Invented models and algorithms to test inheritance and polymorphism relations in object-oriented software (with former PhD student Dr. Roger Alexander)
- **Input validation testing**: Invented a way to filter inputs based on requirements documents (with former PhD student Dr. Jane Hayes)
- **Class integration and test order problem**: Developed new algorithms to solve the CITO problem (with former PhD student Dr. Aynur Abdurazik)

CONSULTANT ACTIVITIES

- Various Software Intellectual Property Cases: TransUnion, TriZetto, National Wooden Pallet and Container Association, Vertel, Creative Labs, Agitar Software, and others

- Technical Consulting and Advising: JP-Morgan, Pyramid Systems, 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
- CS Educational Consulting and Advising: Laureate, Samsung Electronics, United Arab Emirates University, Linköping University

PUBLICATIONS

- BOOKS AND CHAPTERS

  1. *Introduction to Software Testing*, Paul Ammann and **Jeff Offutt**, 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 (*Invited*)

- REFEREED JOURNAL PUBLICATIONS
  (*My students' names are in* **SMALL CAPS** *and my post-doc mentees are in* ***bold***.)

  1. Jeff Offutt and Richard DeMillo. Retrospective on: Constraint-Based Automatic Test Data Generation. IEEE Transactions on Software Engineering, March 2025
  2. Kathiani Elisa de Souzaa, **Fabiano Cutigi Ferrari**, Valter Vieira de Camargoa, Marcio Ribeiro, Jeff Offutt. A Systematic Review of Fault Tolerance Techniques for Smart City Applications. Elsevier's *Journal of Systems & Software*, DOI: j.jss.2024.112249. Volume 219, January 2025
  3. Jamie Colwell, Amy Hutchison, Kristie Gutierrez, Jeff Offutt, and Anya Evmenova. Elementary teachers' perceptions of an online professional development for literacy-focused computer science education. *Computer Science Education*. DOI: 10.1080/08993408.2023.2263831. October 2023
  4. **Fabiano Ferrari**, **VINICIUS DURELLI**, Sten Andler, Jeff Offutt, Mehrdad Saadatmand, Nils Muellner. On Transforming Tests from Models to Code: A Systematic Literature Review. Wiley's *Journal of Software Testing, Verification, and Reliability*. September 2023
  5. **Marcos Lordello Chaim**, **KESINA BARAL**, Jeff Offutt, Mario Concilio Neto, and Roberto Paulo Andrioli de Araujo. On Subsumption Relationships in Data Flow Testing. Wiley's *Journal of Software Testing, Verification, and Reliability*. March 2023
  6. Amy Hutchison, Jamie Colwell, Kristie Gutierrez, Anya Evmenova, Jeff Offutt, Margaret Gross. Evaluating the role of professional development on elementary teachers' knowledge, comfort, and beliefs related to teaching computer science to students with high-incidence disabilities. *Journal of Research on Technology in Education*. July 2022
  7. Amy Hutchison, Jamie Colwell, Kristie Gutierrez, Anya Evmenova, Jeff Offutt, and Valerie Taylor. Designing a model of computer science professional development for elementary educators in Inclusive settings. *Journal of Technology and Teacher Education*, 29(2):165-193, April 2021
  8. Alessandro Viola Pizzoleto, **Fabiano Cutigi Ferrari**, Jeff Offutt, Leo Fernandes, and Marcio Ribeiro. A Systematic Literature Review of Techniques and Metrics to Reduce the Cost of Mutation Testing. Elsevier's *Journal of Systems and Software*, Volume 157, November 2019
  9. **LIN DENG** and Jeff Offutt. Experimental Evaluation of Redundancy in Android Mutation Testing. *International Journal of Software Engineering and Knowledge Engineering*, 28(11), February 2019
  10. **YUN GUO**, **NAN LI**, Jeff Offutt and Ami Motro. Exoneration-based Fault Localization for SQL Predicates. Elsevier's *Journal of Systems and Software*, October, 2018
  11. Jeff Offutt and **SUNITHA THUMMALA**. Testing Concurrent User Behavior of Synchronous Web Applications with Petri Nets. Springer's *Software and Systems Modeling*, February 2018

12. **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
13. **NAN LI** and Jeff Offutt. Test Oracle Strategies for Model-based Testing. IEEE Transactions on Software Engineering, 43(4):372-395, April 2017
14. **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
15. **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
16. 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
17. **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
18. 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*
19. 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
20. 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
21. Jeff Offutt. Putting the Engineering into Software Engineering Education. *IEEE Software*, Jan-Feb 2013, 30(1):96-100. (*Opinion column*)
22. **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
23. **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
24. 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
25. **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
26. **ROGER 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
27. **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
28. 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)
29. 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

30. 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
31. **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
32. 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
33. 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
34. 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
35. **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
36. **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
37. **JANE HAYES** and Jeff Offutt. Input Validation Analysis and Testing. *Empirical Software Engineering*, 11(4):493-522, December 2006
38. Liguo 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
39. **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
40. **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
41. Anneliese Andrews, Jeff Offutt, and **ROGER ALEXANDER**. Testing Web Applications by Modeling with FSMs. *Software Systems and Modeling*, 4(3):326-345, July 2005
42. Liguo 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
43. **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
44. Kai Chen, Stephen R. Schach, Liguo Yu, and Jeff Offutt. Open-Source Change Logs. Kluwer's *Empirical Software Engineering*, 9(3):197-210, September 2004
45. Steve Schach, Bo Jin, Liguo 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
46. 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
47. 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
48. 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
49. 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
50. Jeff Offutt and Shaoying Liu. Generating Test Data from SOFL Specifications. *The Journal of Systems and Software*, 49(1):49–62, December 1999

51. 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
52. 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
53. 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
54. 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
55. 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
56. 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
57. 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
58. 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
59. 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
60. Jeff Offutt and **STEPHEN D. LEE**. An Empirical Evaluation of Weak Mutation. *IEEE Transactions on Software Engineering*, 20(5):337–344, May 1994
61. 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
62. 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
63. 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
64. Jeff Offutt. Investigations of the Software Testing Coupling Effect. *ACM Transactions on Software Engineering Methodology*, 1(1):3–18, January 1992
65. Jeff Offutt. An Integrated Automatic Test Data Generation System. *Journal of Systems Integration*, 1(3):391–409, November 1991
66. Rich DeMillo and Jeff Offutt. Constraint-Based Automatic Test Data Generation, *IEEE Transactions on Software Engineering*, 17(9):900-910, September 1991
67. 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
68. 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 in* **SMALL CAPS** *and my post-doc mentees are in* ***bold***.)

69. **KESINA BARAL**, Jack Johnson, Junayed Mahmud, Sabiha Salma, Mattia Fazzini, Julia Rubin, Jeff Offutt, Kevin Moran. Automating GUI-based Test Oracles for Mobile Apps, IEEE/ACM 21st International Conference on Mining Software Repositories (MSR 2024)
70. Jeff Offutt. Preventing Cheating Is Better Than Curing Cheating, 16th annual International Conference on Education and New Learning Technologies (EDULEARN), Palma, Spain, July 2024

71. Eileen J. Manchester, Brittany Miller, Amy C. Hutchison, Erdogan Kaya, Kelley S. Regan, Kristie S. Gutierrez, Anya S. Evmenova, Jamie Colwell, and Jeff Offutt. Exploring Teachers' Experiences with the Compose and Code Platform for Teaching CS to All Learners American Educational Research Association (AERA) Annual Meeting, Technology as an Agent of Change in Teaching and Learning (TACTL) SIG session, Chicago Illinios, April 2023
72. Amy Hutchison, Jamie Colwell, Kristie Gutierrez, Anya Evmenova, Jeff Offutt, Margaret Gross. Understanding How Elementary Teachers Respond to a Model of Professional Development for Integrating Computer Science Into Instruction. American Educational Research Association (AERA) Annual Meeting, Technology as an Agent of Change in Teaching and Learning (TACTL) SIG session, San Diego CA, April 2022
73. Amy Hutchison, Jamie Colwell, Anya Evmenova, Kristie Gutierrez, and Jeff Offutt. The Effects of the Inclusive Computer Science Model of Professional Development on Teachers Beliefs and Implementation of Computer Science for Students with High-Incidence Disabilities. SITE 2021 - Society for Information Technology & Teacher Education, San Diego, California, April 2022
74. Jeff Offutt and **KESINA BARAL**. Designing Divergent Thinking, Creative Problem Solving Exams. Software Engineering Education and Training (ICSE SEET), Pittsburgh Pennsylvania, USA, May 2022
75. **KESINA BARAL**, Jeff Offutt, Paul Ammann, and RASIKA MOHOD. Practice Makes Better: Quiz Retake Software to Increase Student Learning. 3rd International Workshop on Education through Advanced Software Engineering and Artificial Intelligence (EASEAI), August 2021
76. **Marcos Lordello Chaim**, Mario Concilio Neto, Jeff Offutt. Graph Representation for Data Flow Coverage. 45th IEEE Computers, Software, and Applications Conference (COMPSAC), July 2021
77. **KESINA BARAL**, Jeff Offutt, and **FIZA MULLA**. Self determination: A comprehensive strategy for making automated tests more effective and efficient. 13th IEEE International Conference on Software Testing, Verification, and Validation (ICST), April 2021
78. **Marcos Lordello Chaim**, **KESINA BARAL**, Jeff Offutt, Mario Concilio Neto, Roberto Araujo. Efficiently Finding Data Flow Subsumptions. 13th IEEE International Conference on Software Testing, Verification, and Validation (ICST), April 2021 (Best paper award)
79. **KESINA BARAL**, and Jeff Offutt. An Empirical Analysis of Blind Tests. 12th IEEE International Conference on Software Testing, Verification, and Validation (ICST), Porto, Portugal, April 2020
80. Alessandro Pizzoleto, **Fabiano Ferrari**, Lucas Dallilo, and Jeff Offutt. SiMut: Exploring Program Similarity to Support the Cost Reduction of Mutation Testing. *Sixteenth IEEE Workshop on Mutation Analysis (Mutation 2020)*, Porto, Portugal, April 2020
81. Amy Hutchison, Jamie Colwell, Kristie Gutierrez, Jeff Offutt, Andrew Vardas-Doane, and Valerie Taylor. Examining Elementary Teachers's Perceptions of Computer Science Professional Development Aimed at Integrating Computational Thinking and Coding in Grades K-5 Instruction. SITE 2020 - The Society for Information Technology & Teacher Education, New Orleans, Louisiana, April 2020
82. Jeff Offutt, **BIRGITTA LINDSTOM**, and **KESINA BARAL**. Teaching an International Distributed Discussion-Based Course. The 15th International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS), Las Vegas NV, USA, August 2019
83. **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
84. **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
85. **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

86. **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
87. **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)
88. **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
89. **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
90. **UPSORN PRAPHAMONTRIPONG** and Jeff Offutt. Finding Redundancy in Web Mutation Operators. *Thirteenth IEEE Workshop on Mutation Analysis (Mutation 2017)*, April 2017, Tokyo, Japan
91. 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
92. **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
93. **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
94. **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
95. **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
96. **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
97. **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
98. **BOB KURTZ**, Paul Ammann, and Jeff Offutt. Static Analysis of Mutant Subsumption. *Eleventh IEEE Workshop on Mutation Analysis (Mutation 2015)*, April 2015, Graz, Austria
99. **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
100. **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
101. **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
102. **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
103. **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

104. **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
105. **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
106. 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
107. **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
108. **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
109. **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
110. **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
111. **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
112. **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
113. **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
114. **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
115. **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
116. 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
117. **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
118. **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
119. **JINGYU HU**, **NAN LI**, and Jeff Offutt. An Analysis of OO Mutation Operators. *Seventh Workshop on Mutation Analysis (Mutation 2011)*, March 2011, Berlin, Germany
120. 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

121. **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
122. **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
123. **UPSORN PRAPHAMONTRIPONG** and Jeff Offutt. Applying Mutation Testing to Web Applications. *Sixth Workshop on Mutation Analysis (Mutation 2010)*, April 2010, Paris, France
124. **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
125. **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
126. **SHUANG WANG** and Jeff Offutt. Comparison of Unit-Level Automated Test Generation Tools. *Fifth Workshop on Mutation Analysis (Mutation 2009)*, April 2009, Denver CO
127. **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
128. **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
129. 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
130. **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
131. 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
132. **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
133. **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
134. **MATS GRINDAL** and Jeff Offutt. Input Parameter Modeling for Combination Strategies. *IASTED International Conference on Software Engineering (SE 2007)*, February 2007, Innsbruck, Austria
135. 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
136. **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
137. **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
138. 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

139. 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
140. **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
141. **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
142. **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
143. **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
144. **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
145. **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
146. 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
147. **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
148. 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
149. 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
150. 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
151. Jeff Offutt, **YUAN YANG**, and **JANE HUFFMAN HAYES**. SEEWeb: Making Experimental Artifacts Available. *Workshop on Empirical Research in Software Testing*. Boston, MA, July 2004
152. **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
153. 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
154. 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
155. 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
156. 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
157. **ROGER 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

158. **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

159. **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

160. **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. (**Most influential paper from 30 years of ISSRE**)

161. **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

162. Jeff Offutt. Web Software Applications Quality Attributes. *Quality Engineering in Software Technology (CONQUEST 2002)*, pages 187-198, Nuremberg, Germany, September 2002

163. 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

164. 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

165. 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

166. **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

167. **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

168. 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

169. **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

170. 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.

171. **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

172. **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

173. 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

174. **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

175. **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

176. **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

177. 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

178. **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

179. 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

180. **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

181. **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

182. Jeff Offutt. Software Testing: From Theory to Practice. IEEE AES Systems Magazine, June 1998

183. **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

184. 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

185. 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

186. **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)

187. 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

188. 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

189. 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

190. **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

191. 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

192. Jeff Offutt. Practical Mutation Testing, *Twelfth International Conference on Testing Computer Software*. pages 99–109, Washington, DC, June 1995

193. 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
194. 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
195. 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, pages 473–488, San Antonio, Texas, January 1994
196. **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
197. 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
198. 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
199. 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
200. 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
201. 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
202. Jeff Offutt. Using Mutation Analysis to Test Software. *Seventh International Conference on Testing Computer Software*, pages 65–77, San Francisco, California, June 1990
203. 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
204. Jeff Offutt. An Integrated System for Automatically Generating Test Data. *First International Conference on Systems Integration*, pages 694–701, Morristown, New Jersey, April 1990
205. Jeff Offutt. The Coupling Effect: Fact or Fiction? *Third Software Testing, Analysis, Verification Symposium*, pages 131–140, Key West, Florida, December 1989
206. 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
207. 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
208. 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
209. 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
210. 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

- OTHER PUBLICATIONS

211. **Jeff Offutt**. 56 editorials in Wiley's journal of Software Testing, Verification, and Reliability. March 2007-July 2019.
212. Jeff Offutt. Comments on Tai. *ACM Software Engineering Notes*, January 1990
213. Jeff Offutt. Hints on Writing Style for Usenet. Monthly electronic posting to the Usenet newsgroup *news.announce.newusers*
214. 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 in* SMALL CAPS.)

215. NAN LI and Jeff Offutt, A Test Automation Language for Behavioral Models October 2013, GMU-CS-TR-2013-7
216. Leonard Gallagher and Jeff Offutt, Integration Testing of Object-oriented Components Using FSMS: Theory and Experimental Details, July 2004, ISE-TR-04-04
217. AYNUR ABDURAZIK, Jeff Offutt, and Andrea Baldini, A Controlled Experimental Evaluation of Test Cases Generated from UML Diagrams, May 2004, ISE-TR-04-03
218. Ye Wu, Jeff Offutt, and XIAOCHEN DU, Modeling and Testing Dynamic Aspects of Web-based Applications, March 2004, ISE-TR-04-01
219. Ye Wu and Jeff Offutt, Modeling and Testing Web-based Applications, November 2002, ISE-TR-02-08 http://cs.gmu.edu/∼tr-admin/
220. 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/
221. 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/
222. AYNUR ABDURAZIK and Jeff Offutt, Generating Test Cases from UML Specifications November 1999, ISE-TR-99-09
223. 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
224. 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
225. 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
226. 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
227. 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
228. 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
229. 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
230. 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
231. 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
232. 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

233. 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
234. Jeff Offutt and **S. D. LEE**. IMSCU Programmer's Reference Manual. Clemson University Computer Science Technical Report 91-121, July 1991
235. 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
236. Jeff Offutt and **C. FUNSCH**. Lab Manual 1.0 for Computer Science 241: Data Structures. Clemson University, August 1990
237. 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
238. 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
239. 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

More than three decades of teaching experience at four universities: University at Albany, George Mason University (GMU), Clemson University (CU), and the Georgia Institute of Technology (GIT).

Faculty of the Year, George Mason University Alumni Association, 2020. George Mason University Teaching Excellence Award, Teaching With Technology, 2013. Outstanding Teaching Award, School of Information Technology and Engineering, George Mason University, 2003.

I believe strongly in active learning, collaborative learning, and creative innovations. Introduced many new topics for the fast-growing and young field of software engineering, and invented numerous pedagogical innovations. This includes developing eight completely new courses, substantially revising seven others, and publishing several 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 have been used at universities around the world. He currently leads the *Excellence in Computer Science Education and Learning (ExCSEL)* lab.

- Supervised 19 doctoral students, 2 in progress–11 women, 6 university faculty, 11 senior industry leadership positions, 2 consultants
    - Kesina Baral, *Towards Effective Test Oracle Automation*, December 2022. Software Quality Expert, CQSE GmbH.
    - Yun Guo. *Towards Automatically Localizing and Repairing SQL Faults*, August 2018. Senior Database Engineer at CVent.
    - 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. Co-advised with 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.* Senior Software Engineer at NASA/SAIC.
  ⊙ Recipient of ITEA Fellowship, 2002.
- 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. Consultant at ETRI. (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. Systems Engineering Manager at Harris Corporation.
  ⊙ GMU CS Distinguished Master's Graduate, 1994
  ⊙ Recipient of ITEA Fellowship, 1995.
- Michelle Lee (Li Li), *Object-oriented Change Impact Analysis*. November 1998. Chief Technical Officer, ITHAKA.
- Jane Hayes, *Input Validation Testing: A System Level, Early Lifecycle Technique*. September 1998. Professor, University of Kentucky (retired).
- Roland Untch, *Schema-based Mutation Analysis*, December, 1995. (Co-advised with Dr. M. J. Harrold). Professor, Middle Tennessee State University (retired).

- Supervisor, current doctoral students:
  - Dave Farmer, Concurrent software testing
  - Sharmista Kuri, Web application software testing
- University courses
  - Developed new PhD courses: Advanced Software Testing, Analysis of Software for Testing, Experimental Software Engineering, Special Topics in Web-based Software
  - Developed new MS courses: Software Engineering for the World Wide Web, User Interface Design and Development, Masters Capstone Project
  - Developed new BS courses: Software Engineering Usability Analysis and Design, Design and Implementation of Software for the Web, Software Testing and Maintenance
  - Redesigned: Data Structures *bridge* course, Software Testing, Software Construction
  - Taught many other software engineering and computer science courses

ACADEMIC LEADERSHIP EXPERIENCE

- University leadership roles
  - Chair of CS department (UAlbany)
  - Associate chair for graduate studies 2019–2023 (GMU)
  - Department chair 1998–1999 (GMU)
  - Chair of departmental committees (recruitment, graduate studies, graduate admissions, faculty hiring, P&T, computing infrastructure, web committee) (GMU)
  - Chair of college and university committees (Graduate education working group, Dean search committee, Dean and Chair re-appointment committees, student appeals) (GMU)

  - Member of college and university committees (responsible conduct of research steering committee, graduate council, graduate education working group, research council, P&T, faculty grievance) (GMU)

- Research professional leadership roles
  - Editor-in-Chief of Wiley's journal of Software Testing, Verification, and Reliability, 2006-2019
  - Steering committee chair, IEEE International Conference on Software Testing, Verification, and Validation (ICST), 2007–2011
  - Chairs of various conferences and workshop (General chair, program chair, PhD symposium chair, Workshop chair, etc)

- Curriculum development at UAlbany
  - Led efforts to modernize and streamline three CS programs:
    * BS-CS to increase retention and provide more flexibility in electives by expanding two introductory programming courses into three, and by compressing the number of third- and fourth-year systems courses from five to three
    * MS-CS to modernize and to handle large growth by decreasing the core and increasing electives and by designing concentrations in AI&ML, Systems, and Theory
    * PhD-CS to create growth opportunities by clarifying requirements, reducing the number of required exams, and including enforceable timelines

- Curriculum development at GMU:
  - Led several major curriculum changes in 2021-2023 with the goal of increasing participation in graduate computing programs at George Mason:
    * Redesigned our existing Masters of Science in Information Systems program to be more inclusive, and to be shared between two departments (CS and Information Science and Technology)
    * Designed a new masters program (*Master of Computing*) to implement my vision of a 21st century broad program in computing, which would provide educational opportunities in computing for diverse students from all undergraduate majors
    * Designed a *bridge on-ramp program* in computing as a graduate certficate that enables students from diverse backgrounds to pivot into an MS program in Software Engineering or Computer Science
    * Designed a PhD in CS Higher Education, with the goal of preparing students to instructional positions in university computing departments
    * Designed and unifying multiple Bachelors-Accelerated-Masters programs to enable undergraduate students to overlap up to four courses between their Bachelors and Masters degrees
  - co-Chaired Provost's Graduate Education Working Group, 2019-2020
  - Led a major restructuring of the **PhD in Computer Science**, effective Fall 2018
  - Led two major restructurings of the **MS in Software Engineering**, effective Fall 2018 and Fall 2005
  - Co-led team to create an **undergraduate concentration** in Software Engineering, as part of the Applied Computing Science degree (effective Fall 2009)
  - Led team to create an **undergraduate minor** in Software Engineering (effective Fall 2006)
  - 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
  - Co-led team to design GMU's **PhD in Computer Science**, effective Fall 2000

- Curriculum development at Clemson:
  - Participated in a major restructuring of Clemson University's MS in Computer Science, eliminating requirement for MS thesis (wrote first draft of document)

  - Participated in a major restructuring of Clemson University's PhD in Computer Science (wrote first draft of document)

PROFESSIONAL ACTIVITIES

- Journal Editorial Boards
  - Editor-in-Chief, The Journal of Software Testing, Verification, and Reliability (2006-2019)
  - IEEE Transactions on Software Engineering (2001-2005)
  - The Journal of Software and Systems Modeling (2004-current)
  - Empirical Software Engineering Journal (2006-current)
  - Software Quality Journal (2002-current)
- Conference & Workshop Organizing Committees
  - Co-founder and Founding Chair of Steering Committee, IEEE International Conference on Software Testing, Verification, and Reliability (ICST)
  - Program Chair: TestEd 2020, ICST 2009 (IEEE), Mutation 2007 (IEEE), Workshop on Empirical Studies of Software Maintenance (IEEE), ICECCS 2001 (IEEE)
  - PhD Symposium Chair: ICST 2013, ISSRE 2015, ICST 2017, ICST 2021
  - Workshop Chair, ICST 2019
- More than 150 conference and workshop technical program committees
- Frequent reviewer and panelist for NSF and other agencies
- Frequent reviewer for about a dozen journals
- Professional societies
  - Association for Computing Machinery
  - IEEE Computer Society

ACADEMIC SOFTWARE PROJECT EXPERIENCE

Throughout my career I have focused on developing software engineering techniques based on sound theory and that can be applied to practical situations. I emphasize empirical validations, and as such, have constructed and 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, Godzilla, and Albert) have been used by hundreds of other researchers. muJava and Mothra in particular have been used by thousands of researchers and educators and cited in many hundreds of papers.

- *Mutation Testing for Java (μJava).* 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 Offutt and Yu-Seung Ma at the Korean Advanced Institute for Science and Technology (KAIST), as part of her PhD work. μJava is open source. It has been used by thousands of researchers and teachers and our papers about the tool have been cited well over 1300 times. (*https://github.com/jeffoffutt/muJava*)

- *Coverage web applications.* To enable use of our textbook, Paul Ammann, several students, and I developed several web applications to measure coverage of graphs and logic predicates. Freely available, with source on github, logs show these tools are used hundreds of times per day. (*https://cs.gmu.edu/∼offutt/softwaretest/*)

- *The Mothra tool suite.* Mothra was built in the 1980s and widely used well into the 2000s by educators and researchers in software testing. Mothra was one of the first research tools whose source was distributed widely. As such, it has been called one of the first open-source projects. Offutt and Dr. Rich DeMillo were the primary designers and Offutt implemented more than 50% of Mothra. Offutt and Dr. Kim King designed a special-purpose intermediate code to support mutation, which has been cited as an inspiration for Java bytecode. By using an independent tool architecture approach (much like Unix), designed around shared data structures, Mothra set a new standard for flexibility and adaptability among software engineering research tools.

- *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 generates coverage graphs for Java programs, which can 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 Dr. 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.

- *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, and Rational Corporation's Rose tool. This tool was built by Dr. Aynur Abdurazik as part of her MS thesis.

- *Godzilla.* This tool was designed and built as part of Offutt's doctoral work, and integrated with Mothra. 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. Much of Godzilla's test data generation innovations were incorporated into the commercial testing tool Agitator.

- *HyperMothra, Leonardo, and Equalizer.* These research systems extended Mothra to evaluate efficiency mutation testing innovations. These systems provide extended functionality, use different basic algorithms, and utilize parallel hardware architectures. These projects were all implemented as MS thesis projects, using Offutt's designs.

- *IMSCU.* This streamlined mutation system was built by several graduate students according to Offutt's specification and under his direction. Two versions were built in diifferent programming languages. 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. Albert was used by mathematicians around the world to support research in nonassociative algebra.
https://people.cs.clemson.edu/∼dpj/albertstuff/albert.html

ADDITIONAL STUDENT MENTORING

- Supervised Master's Theses (most resulted in fully refereed published papers as listed above)
  - 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.
- Graduate Student Advisory Committees
  - Ryan Johnson (George Mason, MS SWE, PhD Fall 2019)
  - Bob Kurtz (George Mason, MS SWE, PhD Fall 2018)
  - 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 Brykczinski (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)